Interactive Sonification of Network Traffic to support Cyber Security Situational Awareness

Northumbria University
NEWCASTLE

Mohamed Debashi

A thesis submitted in partial fulfilment of the requirements of the University of Northumbria at Newcastle for the degree of

Doctor of Philosophy

Research undertaken in the
Department of Computer and Information Sciences

October 2018

Abstract

Maintaining situational awareness of what is happening within a computer network is challenging, not least because the behaviour happens within computers and communications networks, but also because data traffic speeds and volumes are beyond human ability to process. Visualisation techniques are widely used to present information about the dynamics of network traffic. 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, visualisations require visual attention and so are not well suited to continuous monitoring scenarios in which network administrators must carry out other tasks. Situational awareness is critical and essential for decision-making in the domain of computer network monitoring where it is vital to be able to identify and recognise network environment behaviours.

This thesis presents SoNSTAR (Sonification of Networks for SiTuational AwaReness), a real-time sonification system to be used in the monitoring of computer networks to support the situational awareness of network administrators. Together with a new way of reducing traffic complexity, called "IP flow", SoNSTAR provides an auditory representation of all the TCP/IP protocol traffic within a network based on the different traffic flows between network hosts. SoNSTAR narrows the gap between network administrators and the cyber environment so they can more quickly recognise and learn about the way the traffic flows within their network behave and change. SoNSTAR raises situational awareness levels for computer network defence by allowing operators to achieve better understanding and performance while imposing less workload compared to visual techniques. SoNSTAR identifies the features of network traffic flows by inspecting the status flags of TCP/IP packet headers. Different combinations of these features define particular traffic events.

These events are mapped to recorded sounds to generate a soundscape that represents the real-time status of the network traffic environment. Listening to the sequence, timing, and loudness of the different sounds within the soundscape allows the network administrator to monitor the network and recognise anomalous behaviour quickly, without having to continuously look at a computer screen. Evaluation showed that operators were able to monitor and recognise network attacks better with SoNSTAR than with Snort, a leading visual intrusion

detection system, and with lower reported cognitive workloads. Accuracy of recognition was highest when using both Snort and SoNSTAR together (97.14%). The results clearly show that accuracy improved when using sonification. When using sonification, the mental and perceptual workloads required were less than when using visualisation alone (45% vs. 58%). The pressure participants felt due to the pace of the monitoring task was less when using SoNSTAR (31% vs. 65%). Frustration rate showed improvement when using SoNSTAR (36% vs. 71%). The very act of listening to the traffic generates a fast discovery process leading to new knowledge of malicious behaviours that is not possible with current algorithmic approaches. SoNSTAR enabled the user to explore distributed, parallel and horizontal behaviours that are similar to normal behaviours. An experiment using the 11.39 GiB ISOT Botnet Dataset, containing labelled botnet traffic data, compared the SoNSTAR system with three leading machine learning-based traffic classifiers in a botnet activity detection test. SoNSTAR demonstrated greater accuracy (99.92%), precision (97.1%) and recall (99.5%) and much lower false positive rates (0.007%) than the other techniques. The knowledge generated about characteristic botnet behaviours could be used in the development of future IDSs.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Dr Paul Vickers for the continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD study. Besides my advisor, I would like to thank Prof. Ahmed Bouridane for his help, care and motivation.

I would like to dedicate this thesis to my teacher and father Mr Mansur Habib-Allah and teacher and grandfather Mr Mohamed Abu-Alanware.

I would like to thank the students of Northumbria University who participated in the evaluation experiments without whom there would be no experimental data.

In addition, I thank my fellow PhD researcher, Dr Mohammad Alauthman for the stimulating discussions and valuable information about network security and the current state of research.

Last but not the least, I would like to thank my family: my parents and my wife and to my brothers and sisters for supporting me, both in practice and spiritually throughout writing this thesis and my life in general.

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others.

Any ethical clearance for the research presented in this thesis has been approved. Approval has been sought and granted by by Research and Business Services at Northumbria University under record number RE-EE-13-140616-539ec5986be50 on 17/02/2015.

I declare that the Word Count of this Thesis is 46,494 words.

Mohamed Debashi
October 2018

Table of contents

List of figures

List of tables

List of acronyms and abbreviations

ACK TCP flag indicates the value in ACKnowledgement is valid (acknowledges received data)

C&C Command and Control

DDoS Distributed Denial of Service

DNS Domain Name System

DPI Deep Packet Inspection

DR Detection Rate

FIN TCP flag indicates that No more data from sender (FINish a connection)

FNR False Negative Rate

FPR False Positive Rate

GUI Graphical User Interface

HTTP Hypertext Transfer Protocol

ICMP Internet Control Message Protocol

IDS Intrusion Detection System

IP Internet Protocol

IRC Internet Relay Chat

OSI Open Systems Interconnection

PCAP A network packet capture library for Python

P2P Peer-To-Peer

PSH TCP flag indicates that PuSH the buffered data to the receiving application

RL Reinforcement Learning

RST TCP flag indicates that ReSeT the connection (Aborts a connection in response to an error)

SYN TCP flag indicates that SYNchronous the connection (Initiates a connection)

SCADA Supervisory Control and Data Acquisition

TNR True Negative Rate

TPR True Positive Rate

UDP User Datagram Protocol

URG TCP flag indicates how much of the data in the segment, counting from the first byte, is URGent

Chapter 1

Introduction

1.1 Background

Technology has changed the way in which computer networks are treated, with national governments increasingly viewing them as a war fighting domain alongside the traditional sea, land, air, and space domains [47]. Cyber domain operations know no national boundaries [79] and understanding where these operations happen is important for a greater appreciation of the cyber environment [158].

The US Department of Defence has defined 'cyberspace operations as a global domain within the information environment consisting of inter-dependent network information technology infrastructures, including the internet, telecommunications networks, computer systems and embedded processors and controllers' [61, p. 126].

Exploring and finding better ways to understand actions and behaviours in the cyber environment could provide network administrators with the advantage of better situational awareness over adversaries and their future attack plans. Cyber operations are complex and can be divided into three main areas [47]:

- **Defence:** These activities represent approximately 80% of the operational spectrum. This includes the entire work done for the purpose of security protection in order to identify adversaries and stop any malicious or attack activities and to maintain information about the current state of the network. These operations include action taken in the network to monitor, analyse, detect, protect and respond to attacks or unauthorised actions [79]. Some organisations even hire white-hat hackers (known as *ethical hackers*) to perform penetration testing and other testing methodologies to ensure the security of an organisation's information systems.

- **Exploitation:** These activities include any preparatory operation conducted to enable future malicious or attack activities. This foot-printing could include port scans or probing to identify vulnerable network components along with other preparations such as spying for the information necessary to make decisions about an attack or injecting a bot or intrusion or trojan into the victim or enemy system or network for the purpose of future intended attacks. These operations include acquiring security passwords or active foot-printing by physical access to victim data-centres or any network component or any building and do not need to be confined to the digital world [6, 79, 151].

- **Attack:** These activities represent the overt phase where attack operations start to affect their targets [158]. This happens when an attacker has identified the target for an attack and made necessary preparation for this purpose such as (FUD, RAT, Root kit and Key loggers) or have acquired security passwords enabling them to perform the intended action. These operations include actions taken to destroy or otherwise incapacitate enemy networks or to steal or change information [6, 30, 36].

While the Internet exposes computer networks to multiple threats from the outside world, it is still considered that internal threats are more dangerous to network security [162]. For most of today's network administrators, it is essential to be able to monitor their network activity in real-time to be able to obtain a current overview of their network environment. Additionally, it is important to be able to quickly adapt to problems as they arise which requires high levels of situational awareness. Computer network operators face new challenges to maintain real-time situational awareness using technology [118]. This research focuses on the first goal of obtaining and maintaining a real-time overview of network behaviour.

Network measurement tools include software and hardware methods to gather data and analyse traffic at different protocol levels. Network traffic analysers underpin real-time data collection and online analysis. The majority of these systems use graphical displays to represent live traffic data. Other measurement software, such as Tcpdump and Wireshark sniff network data in real-time and store it. Once the system has collected and stored the traffic data, it can be analysed offline [163].

1.2 Research motivation

Visualisation has been used as a tool for monitoring networks in order to raise situational awareness levels [47]. 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 [51]. 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 [118]. 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 IDS 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 then have to inspect to try to understand the situation [17, 102]. 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 [44]. 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 visualisation network monitoring and IDS 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 a long period. 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.

Hildebrandt [72] 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 [99, 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, it has to allow administrators 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.

The question this research addresses is how can sonification be used in the maintenance of real-time situational awareness to provide the protocol flow granularity required to understand network environment behaviour? As a solution to this issue network administrators need a real-time monitoring tool to facilitate the acquisition and maintenance of security situational awareness. Such a tool could help with:

- Increasing the understanding of cyber environment which is a vital task for network management and diagnosis.
- Maintenance of network security and situational awareness of malicious events such as intrusions attacks.
- Maintenance of network connectivity and health through monitoring, understanding, and tuning.

1.3 Research aims and objectives

The overall aim of the thesis is to use sonification to support network operators in overcoming situational awareness challenges in computer networks and explore alternative approaches to improve network security threat detection and monitoring in order to raise cyber security situational awareness levels. The approach proposed in this research has the following characteristics. It creates and develops a real-time sonification monitoring system that represents network events using sound based on flow features. Furthermore, it introduces a new type of flow reduces the number of flows required to be sonified to represent the network state. It allows the user to recognise changes in the traffic activities and detect malicious activities as soon as they occur. It provides a demonstration of detecting botnet activity.

In this thesis, the scope is limited to developing SoNSTAR which is a monitoring system as part of a security situational awareness solution. As well as supporting real-time monitoring, SoNSTAR can also read stored traffic datasets in the pycap format for offline review. In addition, SoNSTAR generates log files of its activities to support the learning process.

The main target of the research is TCP/IP network traffic as it represents most of the traffic which passes through computer networks [138, 142].

1.3.1 Objectives

The objectives in support of the overall aim are as follows:

- Study existing network security and monitoring tools.
- Investigate and analyse normal and malicious traffic mechanisms.
- Review the literature on sonification and network sonification research.
- Identify the parameters that might be affected by intrusions and traffic behaviour changes.
- Design and develop a network sonification solution.
- Conduct experiments for testing the proposed solution.
- System validation and results analysis.
- Create a user manual to help people understand how to use the system.
- Train users to use the system and understand the sonification and analyse the outcome.

Fig. 1.1 illustrates the research approach phases. The system development approach used to plan, structure, and control the process of creating and developing SoNSTAR prototyping. Phase 3 in Fig. 1.1 illustrates SoNSTAR framework development steps.

1.4 Thesis contributions

The thesis introduces a sonification monitoring solution called SoNSTAR (Sonification of Networks for SiTuational AwaReness) which includes the protocol flow granularity required to understand network events inside any network environment. We created a new type of

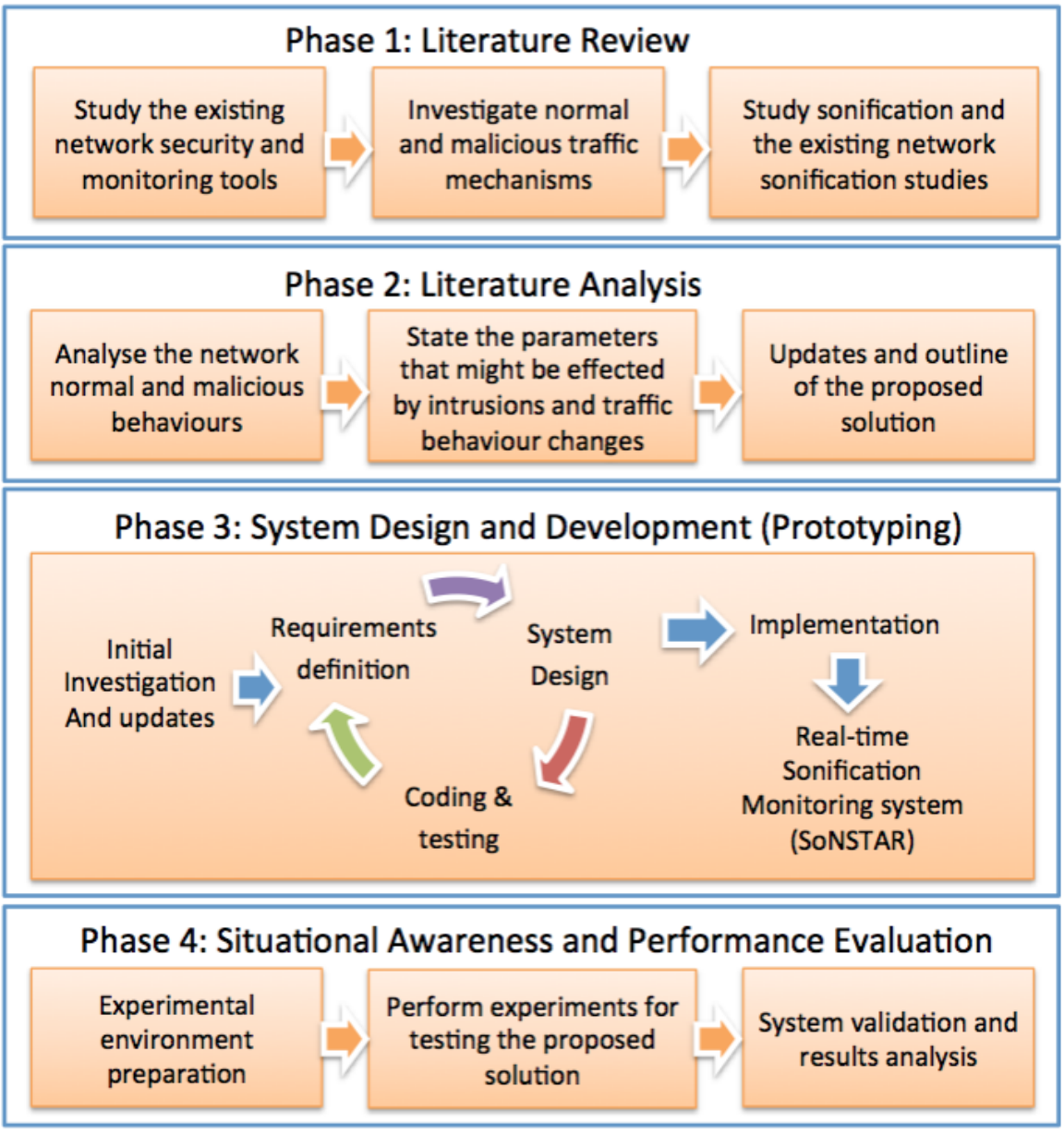

Fig. 1.1 Research Approach

flow, called "IP flow" which is a series of packets and uniquely identified using source and destination IP addresses, and protocol. In addition, the system is enhanced through the use of data reduction methods in order to be able to represent massive volumes of network traffic. It achieves this by using the concept of IP flow to reduce the number of traffic flows requiring sonification, and reducing the number of sounds to be generated by sonification, thus playing the sounds for similar events only once in each time window. Moreover, the system is interactively adjustable and flexible to be behaviour-specific to suit targeted users. As such, the user is capable of tuning the system and establishing thresholds, features and events to suit the user's environment. Furthermore, the system can explore a specific network environment and understand the normal behaviour of that specific network, separate from the generally understood 'normal' behaviours of other networks. This thesis makes three main contributions as follows:

1. The SoNSTAR system itself and the supporting evaluations in Chapters 4 and 5.

2. The introduction of the concept of IP flow which, together with feature construction methods and techniques for representing multiple identical events with a single sound, reduces the complexity of network traffic such that it becomes possible to monitor all the traffic passing through the network.

3. The use of sonification in the discovery of malicious network behaviours, demonstrated in Chapter 6 with a specific case study dealing with botnet activity.

In Chapter 4, a new sonification monitoring solution, called SoNSTAR, is introduced. The work's novelty consists of sonifying in real-time extracted features of network traffic based on the control flag status of packet headers without looking into the payloads. The techniques developed to handle the interaction of the user with the system aim to increase situational awareness levels. SoNSTAR inspects the flag status of each packet in a flow and extracts features by periodically counting each packet type as well as the number of flows, and then uses this information to control the resulting soundscape. The system is complementary to visualisation systems and more situationally informative than visualisation methods, 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 information presented to the user, has not been demonstrated before. The majority of network sonification systems for intrusion detection rely on network metadata extracted from data volumes, packet size and time, source and destination IP addresses and ports or logs generated by network IDS. To approve the new system design, the proposed solution is initially evaluated in terms of the potential of using

packet counts based flag state to represent and understand the cyber environment and to show how packet types are structured into IP flow. The results of the evaluation have demonstrated that the flag's packet counts clearly provide information about the state of the traffic, and that the packet counts and number of flows change according to the traffic behaviour. The number of flows was huge and as such, could not be checked visually in real time. Therefore, the sonification option is essential. Furthermore, the sound design solution is initially evaluated to confirm whether the sounds generated by SoNSTAR are recognisable by users and could be easily comprehended and whether they reduce the workload of the user. The results have shown high detection and accuracy rates and acceptable false positive rates. The NASA-Task Load Index data have also shown encouraging results. SoNSTAR has provided interesting knowledge about network activity that could greatly help users to know how events occur and how network behaviour and flow state change.

Chapter 5 presents an experimental study to evaluate the advantages that SoNSTAR could bring to intrusion detection in order to raise the cyber security situational awareness. The results of the experiment demonstrated the superiority of the proposed sonification approach over using visual monitoring alone, and showed that using sonification with a visual IDS increases cyber security situational awareness. Although the system could be evaluated manually by conducting comparisons against log files, this experiment aimed to evaluate the practicality of using sonification in live monitoring tasks. The results suggest that using SoNSTAR to explore new events and features brings benefits to IDSs and network monitoring in general. Accuracy, precision and workload rates have confirmed that SoNSTAR provides a significant advance in the field of sonification for network security. Furthermore, SoNSTAR evaluation has proven its sonification capabilities as a monitoring tool. Network vulnerabilities can be discovered by SoNSTAR and its user can predict and discover possible attack attempts at an early stage and detect novel attack patterns to allow a zero-day attack. Therefore, users can become aware of the attacks that might target their network, which supports situational awareness.

Finally, in Chapter 6, reports on an experiment to investigate how SoNSTAR could be used to enhance existing IDSs and protect this vulnerable environment of function-specific networks against botnets. Most botnet detection systems perform deep packet inspection (DPI) on the entire network traffic, which requires unencrypted payloads. Other systems target botnet behaviour based on Domain Name System (DNS) traffic. Since the latest distributed denial of service (DDoS) attacks and botnets are capable of launching application layer attacks to evade current detection systems, research reported in the most recent literature advises developing detection techniques based on the application layer of the OSI network model. In this experiment, SoNSTAR, which is based on packets' headers, interactive

sonification and exploring traffic for parallel, distributed, repetitive, horizontal and vertical behaviours events and mapping them into sound, managed to detect all of the targeted behaviours. A technique has been developed to collect more features in order to target botnet behaviours and deal with flow traffic, thus allowing the detection of repetitive flow patterns generated by hosts. In addition, the proposed approach used by SoNSTAR has demonstrated better results compared to a number of existing methods using the same dataset. SoNSTAR was also used to discover new botnet features and patterns that could allow it to help to develop IDSs. Furthermore, these features and events could also be used by IDSs to allow faster detection instead of performing DPI to examine all packets, which represents a huge task for high-speed networks and costs the system significant memory capacity and processing power. Instead, SoNSTAR uses a method to collect selective status information and extract features periodically to free memory and save processing power. These selective features are sonified and provide a real example of using sonification in monitoring to allow users to gather more information about the network environment.

1.5 Thesis overview

This thesis investigates the use of sonification as a monitoring tool for computer network security and situational awareness purposes. It describes situational awareness for the cyber domain and the most commonly used security measures. Furthermore, it discusses the previous work carried out in the field of sonification of computer networks. The thesis then goes on to describe a computer network monitoring sonification system for situational awareness purposes. Experiments to determine whether the system proves useful for cyber security situational awareness purposes are then described and the results are subsequently discussed. The thesis consists of seven chapters, the first being this introduction. The rest of the chapters are organised as follows.

Chapter 2 provides an overview of situational awareness within the cyber environment. It briefly looks into the difference between incident response and real-time analysis of network traffic. It describes the common approaches used in monitoring to provide a sense of network situational awareness. Furthermore, it discusses the background and concepts behind existing computer network security and monitoring systems. The chapter then introduces network protocols and provides a summary of the TCP/IP protocol. Finally, a taxonomy of malicious behaviour related to network traffic is provided.

Chapter 3 introduces sonification concepts and includes a discussion of current sonification approaches and techniques. In addition, the chapter surveys the use of sound in various computer network monitoring studies and applications such as NetSon, InteNtion, Stetho

and self-organised criticality sonification (SOCS). The chapter concludes by describing the different sonification approaches and scenarios and their different data levels of extraction and the state of research of using sonification for cyber security situational awareness and shows where the SoNSTAR system fits within this research field.

Chapter 4 describes the SoNSTAR system that is created for this research as a maint tool for monitoring network traffic and as an additional tool to support existing security settings. It outlines the user specifications and requirements for the system. The chapter concludes by describing the primary design of SoNSTAR and the two primary experiments used to demonstrate the SoNSTAR extracted features and sound mapping design and discussing the results.

Chapter 5 describes an experiment conducted to evaluate the performance of SoNSTAR against the Snort intrusion detection system in real-time. It investigates the performance and situational awareness level that are required to support the purpose of SoNSTAR as a monitoring tool. Moreover, the procedures followed in the experiments and the results are also discussed.

Chapter 6 provides an introduction to using SoNSTAR to recognise botnet behaviours and support existing IDSs to target botnet events based on new features. This is achieved through utilising sonification in order to provide the protocol flow granularity required to understand botnet events inside an environment. Furthermore, the chapter studies networks according to the expected motivations behind an attack and specific network vulnerabilities. It describes new features and four additional algorithms to process them to target parallel, distributed, repetitive, horizontal and vertical behaviours and discusses the extension of the SoNSTAR system and its use in this work to target botnet behaviour. Finally, the procedures followed in the experiments and the results obtained are presented and discussed.

Chapter 7 reviews the material presented in the previous chapters and draws conclusions from the evaluation of the SoNSTAR system and the work presented in the thesis as a whole. Finally, recommendations for further work and research are provided.

Chapter 2

Computer Network Security and Situational Awareness

2.1 Situational awareness (SA)

Endsley defined situational awareness (SA) defined 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' [45, 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 [159, p. 13].

SA allows the network administrator to be aware of a network's current state. This situational perception includes situation recognition and identification [82]. Laing and Vickers [101] explained situational awareness as a process of becoming aware of the immediate environment and grasping how temporal/spatial events (over which one may or may not have control) will affect the environment.

This enables a network administrator to assess network activities for both current and future impacts through understanding the cyber environment, tracking activities of legitimate or adversarial actors. According to Endsley's model [45] SA consists of three phases of recognition, comprehension and projection to reach a resolution. Therefore, malicious activities and the exploitation of the network cyber environment have to be recognised and detected as soon as possible. Security conclusions or resolutions have to be made to prevent future attacks and the misuse of network resources. Other non-successful attack activities or anomalies have to be recognised and identified as quickly as possible in order to acquire more knowledge about the cyber environment.

Boyd's OODA (observe, orient, decide, act) loop theory [4] 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 some features of the environment. The second stage (orient) refers to directing attention towards an adversary. The decide stage involves deciding upon the next actions. Finally, the fourth stage (act) involves implementing the decision. Boyd concluded that American pilots won their battles because they had better training which made them better at deciding and acting [22, 47].

As 'humans are inherently visual beings' [102, p. 65], several visualisation monitoring tools have been introduced to provide a sense of network situational awareness. No system can implement the best security measures without interaction with people. Huge efforts have been made to allow security analysts to observe the current state of their computer networks. However, it is difficult to maintain high SA levels [102]. 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. In this research it is important to discover whether people's listening skills allow them to use a sonification system to recognise network attacks. Everyday life suggests that humans are capable of processing auditory streams to become aware of significant events in their immediate environment through their experience and intelligence, and this might make them capable of using sonification for maintaining SA of sensitive cyber environments [82].

To help improve situational awareness and threat assessment, information fusion techniques have been developed which combine data from multiple sources and include physical sensors and human observers alike [118].

A separate IDS does not have complete information or knowledge to detect intrusions [51]. Integrating evidence of available security systems is the focus of multi-source data

fusion systems, where numerous and varied security controls are joined to deliver accurate situational awareness in the computer network [117]. Offline information monitoring tools might be important but do not support situational awareness because they do not allow the administrator to recognise events and changes in behaviour immediately. A security system using real-time monitoring for situational awareness is essential for providing the administrator with the network current state.

The monitoring solution introduced in this research (SoNSTAR) targets this type of monitoring to support existing security tools, acting as an additional tool aimed at raising SA levels.

2.1.1 Importance of SA

Malicious activities and exploitation of the network cyber environment have to be recognised and detected as soon as possible. Then, plans need to be made and measures put in place to prevent reoccurrences of these attacks and misuse of network resources. Port scans and connection failure in the TCP protocol, for instance, represent a sign of change environment behaviour. These non-success activities or anomalies have to be recognised and identified in order to acquire more knowledge about the cyber environment [82, 142].

Performance forecasts for network resources based on historical performance measurement or real time monitoring information are important to support SA. This monitoring will enable the administrators to characterise and forecast the performance deliverable at the application level from network resources. Such forecasts could be successfully used to help with the maintenance of applications' performance and services. The real-time monitoring of end-to-end quality of service of network and bandwidth is important in the network administrator which will allow control of misuse of the network resources and will keep bandwidth balanced in the network [85, 102, 170]. This view means SA of network activity required to be maintained to ensure an appropriate response to attacks and management of network resources.

The impetus behind the development of these security tools was generated by a number of factors such as:

- The immense technological progress in the domain of network security.
- The necessity to maintain a healthy network 24/7.
- Wider changes in attack methods, motives and behaviours.
- None of the existing security systems can provide the best security measures without the interaction with a human operator [7, 102].

For example, in September and October of 2012, the websites of six banks suffered from a cyber attack which caused days of slowdowns and even a complete break in their services. Security analysts declared the reason was malicious denial-of-service attacks. The reason behind this attack was to overwhelm the banks IT infrastructures in order to stop legitimate customers to use the bank services [37]. This led to a new invention (the "Corero First Line of Defence" device) introduced as a first-line defence before firewalls. Sonification might have helped with the bank's attack problems by providing the administrators of the bank with immediate recognition of malicious behaviours at the point the attack began as well as of the normal behaviour patterns used in this attack to evade detection.

2.2 Network SA and monitoring

Attack and malicious activity volumes in the cyber environment are currently growing at a rapid pace. Security analysts expend large efforts to observe their computer network's current state. However, they are having trouble maintaining high SA [102]. There is a pragmatic need to find acceptable substitutes to the old methods of protecting and monitoring a network in order to have real-time situational awareness influences and stands behind the rapid development of network security tools [118]. Numerous visualisations systems have been developed and designed to increase analysts' security SA in addition to the typical network security systems. The monitoring solution introduced in this research (SoNSTAR) is designed to act as an additional tool to provide current and deep information about the status of the packets moving in the network connections according to flows generated within the network or the internet. Network monitoring today can be roughly split up into two categories as follows.

2.2.1 Types of monitoring

- **Incident response:**

  SA is carried out after network abuse or an attack to assess the damage. For instance, enterprises perform retrospective SA to understand what has happened for forensic investigation or postmortem analysis [30].

  Most of the available tools on the market use log files. Many types of router log schemes exist, but most of them use the Cisco NetFlow logs or have much in common with NetFlow. NetFlow aggregates network traffic by transforming packet info into flows. Flows are uniquely identified using source and destination IP addresses, source and destination ports, and the protocol, input interface and TOS (type of service) fields.

The accumulated information about the flow is recorded in the router. However, each router has a cache where the flow records are stored. Because of the high speeds of the traffic involved, the storage mechanism only samples packets and not all the flows are preserved [34].

Incident response is concerned with investigating something that already has happened, and about which nothing can be done. The reason behind such analysis is to find evidence of what has happened. An incident response check allows the administrator to go through the data several times and examine any suspicious and interesting flows which are found. It is used to analyse an attack, identify details and provide reports and results which might be presented as evidence in a court of law [30].

- **Real-time monitoring:**

  A real-time monitoring system is a monitoring system where the accuracy of the system behaviour depends not just on the logical outcomes of the computations but also on the physical time when these outcomes are produced [94].

  It is essential for administrators to recognise the traffic volume of their network and the state of the network components [159]. Real-time monitoring is vital to conduct live network traffic analysis to distinguish normal and anomalous behaviours. An environment that supports SA should allow a network administrator to quickly evaluate high-level information such as the cause of an attack [45]. SoNSTAR is a real time monitoring system. Real-time monitoring is an ongoing process, which returns information about the network states with a low latency so that the administrator can respond to the attacks.

  Real time cyber SA aims to collect information about the current state of the network in order to have perception. An SA system provides information about the current state, attributes and dynamics those related to the component of the environment. SA takes in classifying information into an understood picture and provides the basis for comprehension and projection [82].

2.3 Typical network security systems

Network security tools, monitoring systems, and sensors are usually installed at the network gateway. In addition, monitoring software might be installed to provide extra information about network traffic. Fig. 2.1 illustrates a router that provides internet connection to a network. For the most part, a network's first-line safeguard is the firewall, which can be

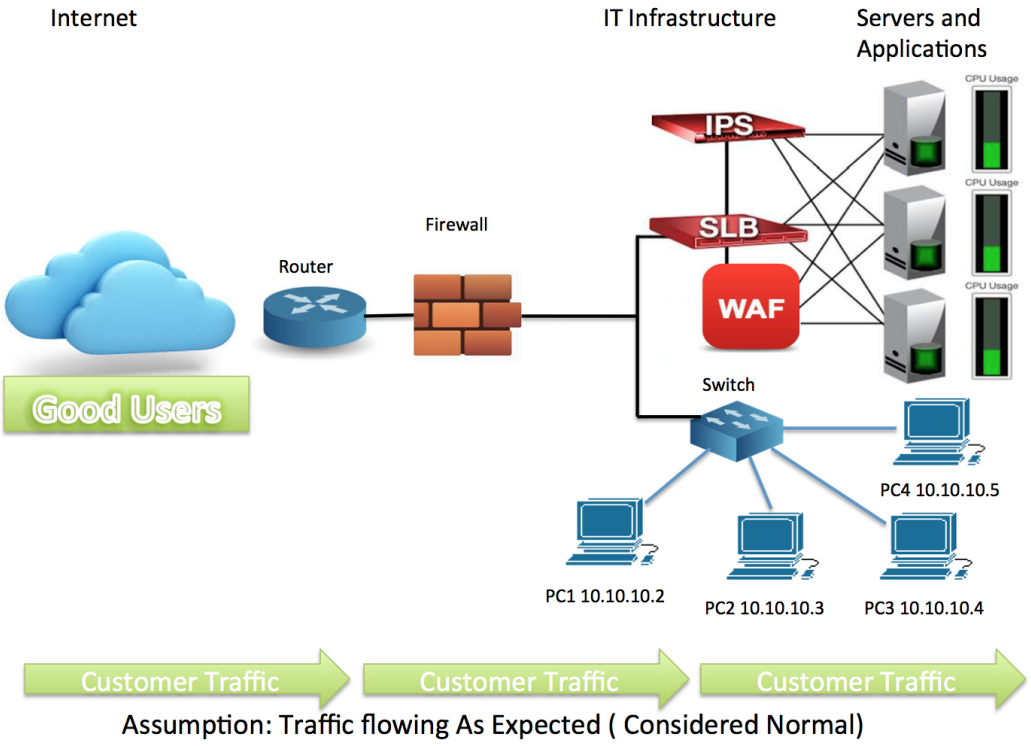

Fig. 2.1 Illustration of a typical network topology (based on [37])

software or hardware based which goes about as a network traffic filter by analysing data packets to prevent or to permit traffic based on a set of rules. Firewalls are most commonly installed at the network gateway. Web application firewalls (WAF) might also be installed. WAFs are designed to keep the servers and applications safe and performing well and, because of that, many successful attacks nowadays are not at the network level, but at the application level. However, attacks still have an effect on the network layer [17].

Other security gadgets can be installed behind the firewall. Intrusion detection systems (IDS) are considered the second line defence. Network administrators use IDS software or sensors that inspect traffic on the network and wait for malicious events to occur. A wide range of monitoring tools is available for users to raise security levels and analyse and monitor the network environment behaviours internally which is considered the third or fourth line defence [17].

Intrusions are defined as 'attempts to compromise confidentiality, integrity, or availability of data, or to bypass the security mechanisms of an IT system' [2, p.147]. An IDS monitors the network assets in order to detect misuse or anomalous behaviour. Intrusion prevention systems (IPS) are similar to IDS but with small extra functions. An antivirus program is likewise fundamental, protecting against malicious software, including classic viruses, RATs,

malware, worms, zombies and trojan horses. Anti malware software and spyware utilities ought likewise to be considered [29, 140].

2.4 IDS technologies

IDSs are classified as active and passive, and come in two broad types: network-based (NIDS) and host-based (HIDS) [44]. An active IDS is configured to provide real time response to attacks and automatically block them, while a passive IDS is configured to provide real time monitoring of network traffic activity and to send alerts to an operator according to any vulnerability and attack. Network intrusion detection systems use a network appliance with a network interface card, while host intrusion detection systems are usually installed, for example, on a server or workstation that is intended to be monitored. A host-based IDS can only monitor individual servers or workstations using software applications and cannot monitor the whole network [44].

2.4.1 IDS detection methods

Most detection methods (especially IDS) depend on packet headers, the payload, or a combination of both to detect attacks and malicious activity. In anomaly-based systems (ABS) analysis of the packet's payload is used to differentiate between normal traffic and anomalous activity. Signature-based systems (SBS) rely on matching patterns against 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 [44, 144]. While anomaly-based systems can detect novel attacks they generate more false positive results and so risk blocking legitimate activity. Anomaly detection methods can only be as good as the anomalies they were programmed to detect. IDSs classify traffic activities as normal or anomalous and an alarm is triggered for anomalous events. Alerts generated are saved into a log file which can be accessed by the operator. Dealing with IDS logs could be used to investigate false positives depending on how much time the operator can invest and how much security he or she needs [44, 125]. 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 [3, 18, 145].

These ways work to an extent, however, they can be defeated by a sophisticated attacker, For example, the attacker crafting the traffic may have access to the same IDS tools that the target network is using, and may be able to create a bespoke attack to specifically avoid their security measures. Unless the network administrator creates their own unique baseline of specific network traffic according to it is functionality, the IDS will continue to fail to detect attacks or might generate more false positive which could be costly in both time and money to maintain.

2.5 Monitoring systems detection methods

Several types of monitoring systems use network usage patterns for detection, measuring 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 [20]. Detection methods can take different approaches depending on the system detection targets. Some use statistical methods to model data according to its statistical properties and use this information to estimate whether a test sample represents the same distribution or not [109]. Some systems train neural networks for detection and classification based on previously detected features and malicious patterns [88]. Other modern detection systems involve reinforcement learning techniques. Here, the neural network is used with a learning algorithm as a classification technique, which showed powerful capabilities. In addition, utilising a reinforcement learning algorithm improves the capability of the detection system [122, 153].

2.6 Network traffic measurement

Network traffic measurements provide essential information for network administrators to support network management. Network traffic measurement is the process of measuring the amount and type of traffic on a specific network. This is very important with respect to effective bandwidth management. Collecting information about the packets transmitted over the network including their time info, IP address, contents etc allows these detailed packet level measurements to be used to monitor network and user behaviour and network traffic distribution [49, 163].

2.6.1 Types of measurement

Network measurement infrastructures support two types of measurement data collection techniques: active and passive measurement. Active measurements require injecting test packets into the network traffic. Passive measurement collects data directly from the network [27].

2.6.2 Network measurement and monitoring tools

Network measurement tools include hardware and software approaches. Measurement tools collect data and analyse traffic at different protocol levels. Many network traffic analysers support multilayer protocol analysis. Network traffic analysers support real-time data collection and online analysis, and the majority of these systems use graphical displays to represent live traffic data. Other measurement software, such as tcpdump and Wireshark, collect and store network data in real-time after which it can be analysed offline [163]. The majority of measurement tools include [81, 85, 93, 102, 170]:

- a user interface (e.g., web, graphical, console);
- online and/or offline traffic graph representation, monitoring network behaviour and reporting against predefined rulesets;
- identification of sender and receiver IP addresses (both local and remote), port numbers, protocols, time, and bandwidth quotas;
- presentation of information as static visualisations (graphic or tabular) of traffic totals and subtotals traffic, and often support traffic shaping, content filtering, usually release alarm and notification;
- support for network mapping and discovery.

2.7 Security systems functionality and limitations

Security systems are complex, and what one system can detect another may not. Any type-based approach (e.g., header-based, payload-based) cannot represent a solution to detect and monitor modern attacks. For example, header-based systems are appropriate to detect attacks and threats directed at vulnerabilities in the network and transport layers and other probing attempts that are used before the attack happens. On the other hand, payload-based systems are more suitable for attacks at the application level which have started gaining importance

due to the increase of web-based services (e.g., electronic government and electronic business systems). However, payload-based systems have an issue with application protocols such as SSH and SSL because of encryption. A possible solution is to use a host-based system to access data after decryption but this also has a problem of causing overhead on the monitored host. Moreover, this problem will become more important when IPv6 starts replacing IPv4 because it uses cryptography to adds authentication and confidentiality to the packets [44].

Most of the real-time monitoring tools currently available focus on connection and traffic statistics. For example, most of them capture the network traffic in real time, analyse it, and produce statics such as bandwidth usage and display the data received and sent by every host in a network in various ways. They try to provide the network administrator with numbers and graphs representing traffic sizes (measured in bytes) from specific categories such as total traffic sent and received and the number of alarms, the number of IP addresses, the number of current connections, the number of active devices, the number of offline devices, the number of internal and external connections and the usage of downloads and uploads of each user. However, this information might be important but it does not allow the administrator to recognise novel events and behaviour changes. There is a need to develop real-time monitoring systems capable of showing changes as they happen through packet level inspection and giving an indication to the administrator about immediate events and behaviours within the traffic [2, 44, 85, 88, 153, 170].

2.8 Network traffic

Before looking at the malicious network attacks, we will briefly look at some of the essential information about network traffic and Internet protocols.

It is important to monitor TCP packets because they comprise most of a network's traffic. TCP traffic can be monitored because TCP packets carry a state flag which show each packet's function [138]. It is more difficult to monitor the state and functionality of UDP traffic because the UDP protocol is connectionless; there is no handshake mechanism to establish a communication channel and so there is no mechanism to guarantee the delivery of packets, their reception in the correct order, or the prevention of packet duplication. For DNS traffic, it would be a good idea to start by monitoring DNS server response errors to a queried domain because it is designed to respond with answers to the queries against its DNS domain records database. ICMP traffic has some complications because it supports the other protocols by generating responses to errors with IP operations. ICMP errors are routed to the source IP address of the packet [48, 95, 136, 142].

Table 2.1 List of monitored failure types [136]

Protocol	Type	Description of the failure
TCP	i	TCP SYN sent, but got TCP resets (RSTs)
	i	TCP SYN sent, but got ICMP unreachable
	t	TCP SYN sent, but timed out
UDP	i	UDP sent, but got ICMP unreachable
	t	UDP sent, but timed out
DNS	i	A DNS server sends error response to a queried domain

There are different ways hosts can respond (or fail to respond) leading to a failure to establish a connection and these should be considered when dealing with network traffic. Table 2.1 shows two types of failure. Type (i) failures can be detected immediately and the timeout, Type (t), failures need time to be detected [136].

2.8.1 TCP/IP introduction

Transmission Control Protocol/Internet Protocol, or TCP/IP, is part of the Internet protocol suite. As this protocol plays a central role in network traffic, a brief overview is now given. In the network layer of the Internet, the most common datagram type used is Internet Protocol version 4 (IPv4), though IPv6 is now experiencing increased usage [95]. Fig. 2.2 shows the structure of an IPv4 datagram.

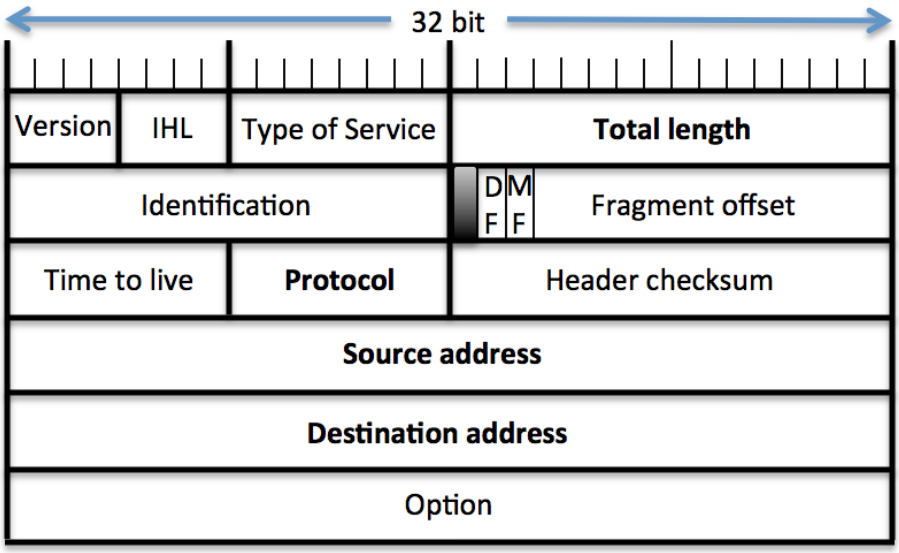

Fig. 2.2 Illustration of the structure of an IPv4 datagram header (based on [95]).

IP datagrams are constructed by a source host which also allows it to insert false identity information, a practice known as “spoofing”. The most common IP datagram protocols are TCP, UDP and ICMP [48, 136].

These protocols are divided into two classes: connection oriented and connectionless. A connectionless protocol only writes a destination in the packet and despatches it with no checking to see if packets have arrived at their destination or if they arrived in the correct order. Familiar connectionless protocols are UDP and ICMP [48].

A connection oriented protocol ensures that the packets arrive in the right order. TCP is such a protocol. The TCP datagram header has various counters and flags which are used to maintain the state of the connection. The flags play a central role in both network denial of service attacks and probes [48].

The TCP header has six flags: URG, ACK, PSH, RST, SYN, and FIN. One of the significant roles that flags play in the TCP protocol is to start and close connections. A mechanism known as the three-way hand shake is required to establish a TCP connection between source and destination and is illustrated in Fig. 2.3 where the initiating source is the client and the destination is a server. The first step in the process is the client host sends the server a TCP packet with its SYN flag set. When the server receives the connection request packet it will allocate some resources for handling the connection. Next, the server host replies to the client with a packet in which the SYN–ACK flag is set to indicate acceptance of the connection. The third message is the client sending the server a packet with ACK flag set indicating acknowledgment of agreement. In this way, the hand shake process is successfully finished and a connection (also known as a “socket”) is established. If the destination is unable or unwilling to establish a connection with the source, it will send back a packet with either the RST–ACK flag set, or an “ICMP Port Unreachable” packet [95].

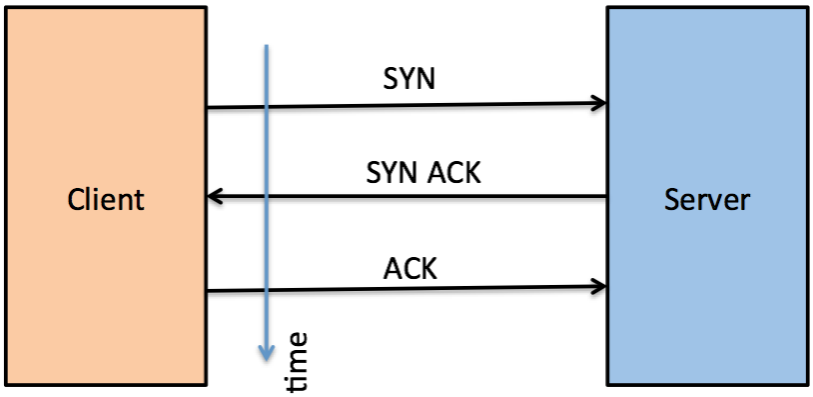

Fig. 2.3 Illustration of the three way handshake (based on [95]).

After the exchange of data and at the end of the connection either side will terminate the connection by using a four-way handshake, where both source and destination send two packets with the ACK and FIN flags set. A normal TCP flow will therefore always contain at least ACK, SYN, and FIN packets in both directions [48, 95, 136, 142]. All of this process is controlled by TCP flags. The status of these flags carries the important information about a packet's role in the connection.

2.9 Malicious network activities

A majority of networks today rely on security systems which have been installed at the gateway, such as firewalls and IDSs. However, network administrators still use other supporting monitoring systems to allow them to understand more about their network environment. While the firewalls and IDSs protect and monitor the network from external threats, the other tools monitor the network for issues, intrusions, and attacks from inside. Moreover, they monitor problems resulting from overloaded and crashed servers. Also, problems related to traffic distribution and connections in the network and the attacks generated from inside the local network are not seen as malicious and are hidden to IDSs. Therefore, administrators use a combination of tools in order to monitor their network [44, 142].

Firewalls, IDS, IPS, and packet filtering devices have limited capabilities to detect and block activity. Some attackers depend on volumetric attacks (e.g., SYN flood and HTTP GET floods) and these systems are not designed to deal with them, they just simply freeze up. Other attacks might be "low and slow", employing small traffic volumes which may appear to be completely normal but which put pressure on the infrastructure leading to network devices experiencing strain due to the volume of requests leading, potentially, to failure (as in distributed denial of service attacks). Some of these security devices do not support packet reassembly where the signature is split over multiple datagrams and some attacks use highly fragmented traffic to exhaust system resources [114].

2.9.1 Categories of intrusion

To increase human understanding of malicious attacks taxonomies have been developed to categorise intrusions, the most widely used being the DARPA evaluation taxonomy which divides intrusions into four groups [104]:

- **Probing (PRB):** Generally used to gain information about the host. (e.g., port-scan, ping) where an attacker scans a network or host to gather information or to discover known vulnerabilities.

- **Denial of service (DoS):** Make a service unavailable to other users. (e.g., ping-of-death, SYN flood, DDoS).
- **Remote to Local (R2L) attacks:** Unauthorised access from a remote machine gaining illegal access to a local account or service. Usually attackers try to gain access to a remote machine by sending packets to a network to generate a vulnerability on that machine to allow them to gain access as a local user of that machine.
- **User to Root (U2R) attacks:** An unprivileged user gains super user rights through a vulnerability illegally granting root access.

Probes, DoS, and DDoS attacks can be detected from the behaviour of network traffic. It is difficult, but not impossible, to detect R2L and U2R attacks because they happen at the application layer in the OSI model. It is possible to get indications that R2L or U2R attacks are occurring. For example, if a remote attacker were trying several passwords through SSH, one would see many short connections on port 22 on the victim's machine. A U2R attack might be detected if the attacker added and started new services on the victim's machine or server [103, 142].

Although these malicious intrusions belong to different categories, they are interconnected and some attacks and malicious activities could contain two or more of them. It is important to Internet service providers (ISP) and web service providers (WSP) to detect both probes and DoS attacks immediately to identify attackers. Identifying DoS and DDoS attacks in the early stage is important to maintain network service. These types of attacks are more easily detected by the ISP or WSP than by the clients. Probes are also important for them, as this provides information about possible future attacks from the hosts where the probing attacks originate [103, 142].

2.9.2 Probes

The purpose of a probing attack is to discover the vulnerabilities and available services on one or more target hosts. On a computer network, attackers use probes to collect information about a network prior to the main attack. The attacker uses the information gained from a probe to identify the operating system (OS) and applications running on the targets and then proceeds to exploit their weaknesses. Therefore, it is important to study this kind of malicious traffic. Detecting probes provides indications of attacker intent and what they are interested in and what hosts are expected to be targeted in future attack plans [58].

A probe is commonly called a 'scan', and they are classified according to the technique used to perform them [58]:

- **Host scan**: One or more ports are scanned on a single host.
- **Port scan**: One or more ports are scanned on more than one host.

There is an expanding requirement for effective security monitoring systems to detect and stop illegal accesses. Common intrusion techniques start with attackers probing a set of hosts and ports to explore any open ports or locate vulnerable servers on a network. This behaviour aims to gain unauthorised access to a target host (R2L). This is often followed by more damaging attacks and exploits. Detecting these types of scans is very important to identify sources of intrusion before any malicious attacks take place. The identification of scanning sources could lead to exposing other types of suspicious behaviour. Therefore, the port scan is seen as the first step of an attack [52, 83].

Port scan

Computer ports provide virtual data connections that can be used by programs to exchange data directly. The most frequently used transport layer protocols are TCP and the user datagram protocol (UDP), in which each packet header has specific source and destination port numbers. Some applications use specifically reserved ports numbers for receiving service requests from client hosts. The well-known ports are defined and assigned by the Internet Assigned Numbers Authority (IANA). On most systems these ports can be used only by system processes or programs executed by privileged users [38].

Scanning a port on a target machine can have one of two results, either the port is open or closed. The flows between the attacker's machine and the victim's machines will be different for each situation. Therefore, for each scan type, the flow is expected to take the behaviour of at least two patterns. If there is no host at the IP address being scanned there will be only the flows coming from the attacker and no response will be received from the target [52].

Probes most commonly use the TCP protocol as it is connection oriented. However, there are connectionless probes using, for example, UDP and ICMP. The most well-known scanning programs are Nmap, Ipsweep, Mscan, Satan, SAINT, Hping3 and portsweep [112]. Port scanners can be classified into two categories [52]:

- **Brute force scanners:** In a brute force scan, all the ports in a specified range on the target machine are scanned sequentially.
- **Stealth scanners:** Stealth scans do not establish a full connection with the victim's machine. Instead, they send one packet with a specific flag set to the targeted ports and based on the replies it can be understood whether the ports are open or not [52].

Stealth scanning is a wide term. However, it is commonly associated with scans that avoid detection. Avoiding detection is subject to the current state of technology [52]. Therefore, today's stealth scans nowadays might not be so classified in future. For example, the Nmap tool performs three types of scans: Null, stealth FIN, and Tree scan, which used to be considered stealthy but, because they are now easily recognisable, they are no longer considered to be stealth scans.

The idea behind these port scans is that closed ports are required by RFC 793 to reply to these packets with "ACK RST". Therefore, if a port does not respond, it would be considered open or the machine does not exist. Machines running the Windows OS machine do not conform to RFC 793 at this point, and will not respond to these packets [52, 58].

Scans take different types and patterns as follows [52, 58, 77, 112]:

- **TCP connect scan:**

  Also, called open scanning, a TCP connect scane is performed using a full three-way TCP/IP handshakefor each of the targeted ports. In this type of scan, a huge number of flow connections on several ports are established on the victim's machine. The establishment of the flow connection at a port means that the corresponding port is open.

  If the scanner identifies an open port, generally it will receive a SYN-ACK, PSH, or FIN packet from the victim's machine, depending on the service running on the scanned port. The scanning machine might terminate the connection by sending an RST to the victim's machine.

  Once a connection to the scanned ports has been established and the open ports identified, the connection is terminated. Since a full connection has been established during the scanning process, TCP protocol options such as sequence and timestamp can be used for attack detection purposes.Thus, if a certain host establishes a large number of connections with multiple ports in a very short period of time, it can be inferred that a TCP connect scan is incoming from that particular host.

- **SYN scans:** The SYN stealth scan, also known as the "half-open scan" is linked to the TCP connect scan. Here, a large number of SYN packets is sent from the scanning machine to the destination host. When they arrive at open ports, if the victim's machine replies with a SYN-ACK packet to accept the connection, then it means the target port is open. The scanning machine does not complete the connection establishment handshake process and ends the connection. If the SYN packet arrives at a closed port,

the victim's machine replies with an RST SYN packet. Often, the SYN scan sends only two packets to each scanned port. This scan type can be quickly recognised if there is a large number of SYN packets incoming from a particular host.

- **ACK scan:** In this type of scan, a number of ACK packets arrive at the victim's machine. Usually, this scan is carried out in order to map firewall settings. The attacker sends an ACK packet with random acknowledgement and sequence numbers to particular ports. If a reply of RST comes back, this port is considered as "unfiltered". If there is no reply or ICMP error is received then that port will be considered "filtered". This type of scan is used to detect filtered services and the type of firewall on the victims network.

  This scan type can be quickly detected if there is a high number of ACK packets incoming from a particular host. Also, It can be identified even in the case it is incoming from multiple hosts when there is a sudden increase in incoming packets with the ACK flag set and corresponding RST replies.

- **FIN scan:** In a FIN scan a larger number of FIN packets arrive at the victim machine. If the victim responds with an RST packet, it means that the port is closed as open ports directly ignore FIN packets. This type of scan can be immediately detected if there is a high number of FIN packets incoming from a particular host.

  This scan type bypasses the traditional network filters and firewalls. The attack's purpose is to recognise active services and the filtered and unfiltered ports in the hosts.

- **NULL scan:** In this type of scan, a high number of packets with no flags set arrive at the victim machine. If a reply of RST comes back, the port in question is considered as close. If reply is received then the port is considered open as open ports ignore such null packets. This type of scan can be simply recognised if there are a high number of null packets incoming from a particular host.

- **XMAS scan:** In the XMAS scan the packet flags FIN, PSH, and URG are set. If a reply of RST'packet comes back, the port is considered closed. If there is no reply the the port is considered open as open ports ignore these packets. This type of scan can be simply recognised if there are a high number of packets with the FIN, PSH and URG flag set incoming from a particular host.

- **UDP scan:** In this type of scan, a high number of zero-byte UDP packets arrive at the victim machine. This scan identifies which services are running on the victim's machine. This attack's purpose is to recognise vulnerable UDP services on the victim's

network. The attack could be discovered by watching for multiple zero-byte UDP packets within a particular time window.

- **ICMP scan:** The scan happens by sending ICMP echo requests to particular hosts to see if hose particular host machines are on or off.

- **Fragmentation attack:** In this type of scan the attacker bypasses the rules set by firewalls by splitting packets into small fragments and sending each part individually. The packets pass the firewall as rules intended to block individual parts are not available.

  This can be identified if there is a high number of packets with a header containing strings that are too short.

Flow characteristics of a scan

The characteristics of a traffic flow depend on the type of port scan used. However, there are some common characteristics as follows.

- Inexpected increase in the number of one or two packet message types in a connection.
- Flows are small and do not contain much data transferred between hosts.
- Flow characteristics are different for open and closed ports.
- The RST packet is often sent as a reply when a port is closed.
- Packets are incoming with particular flags set which identify their purpose.

Port scans exhibit the following specific characteristics:

- Many packets incoming from a particular host to many target hosts.
- The same set of ports is sequentially scanned for each different target host.
- The victim machine receives many flows at different ports.
- A sudden increase in the number of incoming flows into the victim machine.

2.9.3 Denial of service

A denial of service attack (DoS) is 'an explicit attempt by attackers to prevent legitimate users of a service from using that service' [84, p. 2]. The main reason for this type of attack is the exhaustion of system resources to make the service unavailable by exploiting system vulnerabilities or consuming the traffic connection bandwidth.

Some types of DoS attacks use hidden software called "bug" to exploit the victim's device and to disable the service. This type of exploitation cannot be discovered from the TCP/IP packet header unless the software bug is in the network or the transport layer. Often, these bugs use a higher layer than the network layer (such as the application layer) to perform the exploit. It is difficult to identify a DoS attack based on software exploits from the traffic flows because they behave as normal flows. However, it could be identified from the behaviour change of the requested services [58, 112]. For example, a sudden increase in a server load. Fig. 2.4 shows a classification of DoS attacks.

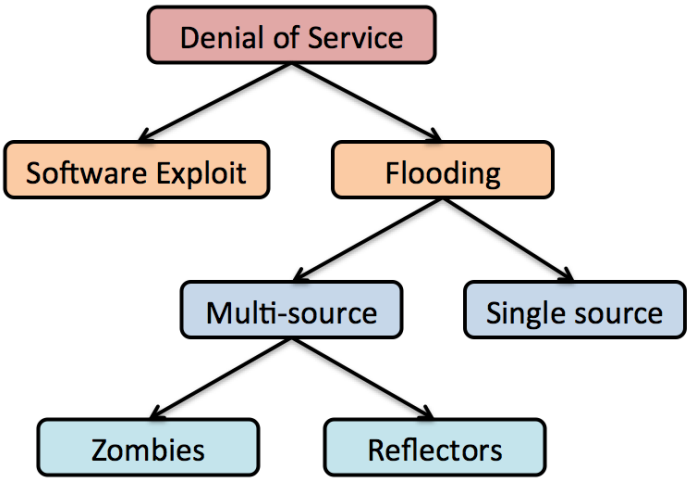

Fig. 2.4 Illustration of Classification of Denial of Service attacks.

Flooding is another technique used to perform DoS attacks. This technique aims to overwhelm the victim machine with traffic by consuming its bandwidth so that service provided by the victim machine is damaged or useless. This type of attack could be detected based on its characteristics which are different from normal traffic. The flooding attack has two forms, single called DoS and multi-sourced called DDoS. DoS is a single source attack requiring the sending of packets directly from the attacker machine to the victim machine. DDoS is a multi-sourced attack requiring the sending of packets from multiple attacker-controlled machines to the victim machine. DDoS attacks are performed using botnets. A single-sourced attack can look like it is incoming from several sources by spoofing the source IP addresses [58, 77].

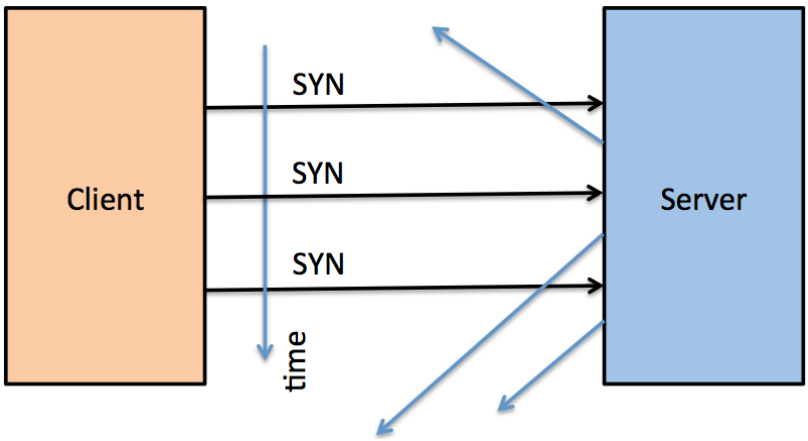

Fig. 2.5 Illustration of a reflection attack using the TCP three-way handshake by spoofing the source addresses.

Distributed denial-of-service (DDoS)

DDoS is a multi-sourced attack in which many machines are coordinated to launch an attack on victim machines. To start this attack, the attacker has to have a control over some compromised machines on the Internet (called "zombies") or make other machines on the Internet send packets to the victim machine as replies to spoofed packets (called "reflectors") [58, 77] (see Figs. 2.5, 2.6).

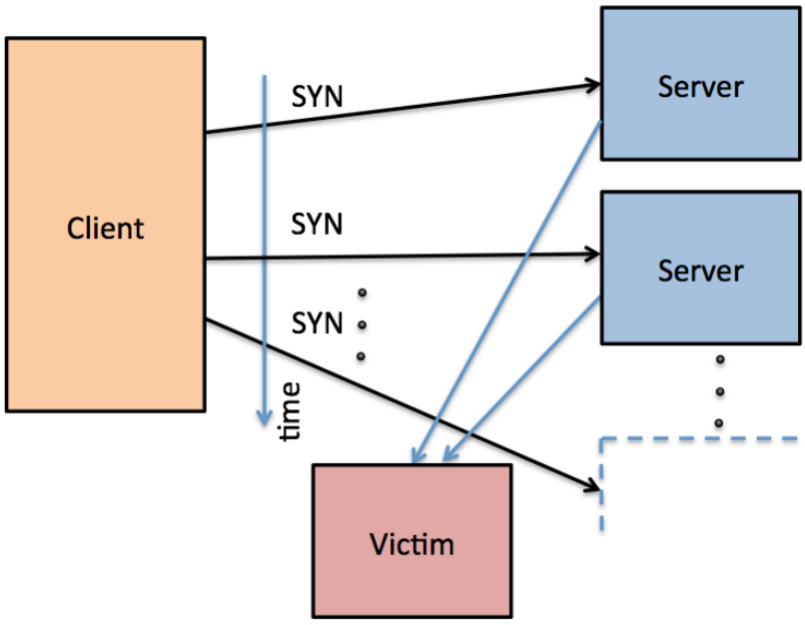

Fig. 2.6 Illustration of a reflection attack on a single victim machine using multiple servers.

DDoS and other advanced cyber-attacks are becoming more common against every type of organisation and institution with a public web presence. They are easy to carry out because there is no requirement to actually penetrate the network as their goal is simply to prevent legitimate users from accessing web services. A related issue is where business competitors are continuously visiting their website to gather information on web pages to compare prices to stay competitive. Although this is not denial-of-service attack it is still traffic that consumes IT resources. Unless the administrators have a clear idea about what is going on in their network and perform an action to prevent misuse of resources and to stop any damage, the situation might become worse and the network could suffer outages [58, 112].

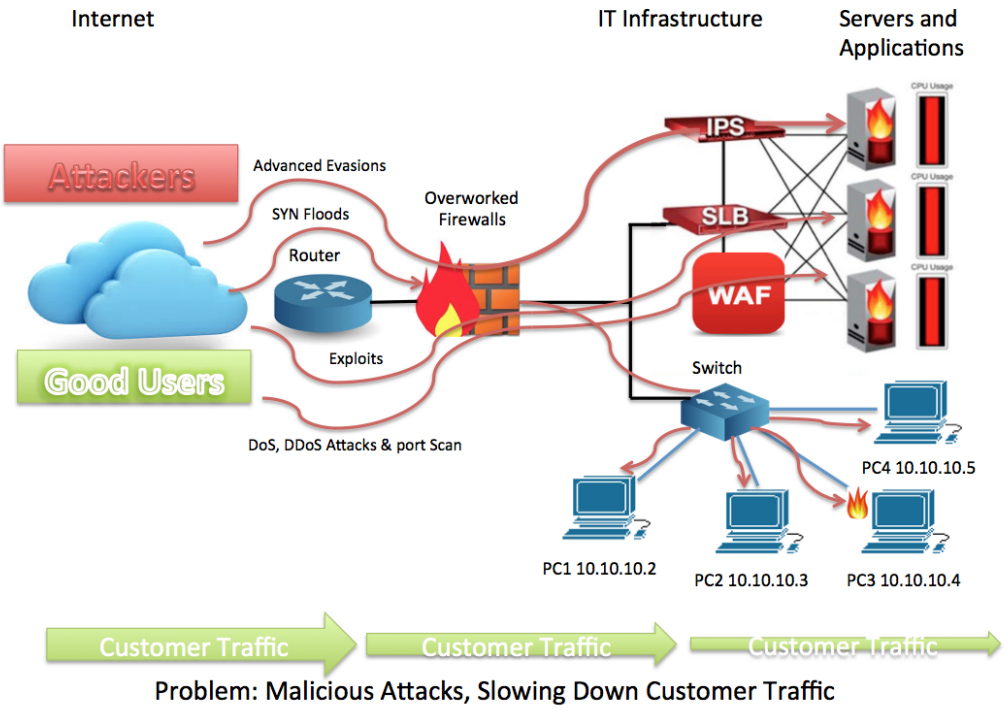

Fig. 2.7 Illustration of a typical network topology under attack.

Fig. 2.7 illustrates a typical network topology under attack using volumetric or other attacks, including advanced evasions, SYN floods, server side exploits and other low and slow application layer DDoS attacks [37]. The most popular DoS attacks are [58, 112]:

- **Land:** In this DoS attack, the attacker sends SYN packets which have been manipulated. The source and destination IP address are the same. Often this attack is performed against badly configured networks and uses zombies for carrying them out.

Such an attack could be identified by recognising packets with the same source and destination IP addresses.

- **SYN flood (Neptune):** This attack uses a half open TCP connection. The attacker floods the victim with SYN packets and receives SYN ACK replies but does not complete the connection with ACK. These requests cause denial of access to legitimate user requests and causes victim server or machine failure. This attack could be recognised by the increase of SYN packets coming from the same host in short time window.

- **Process table:** This attack is often used against Unix systems where an attacker tries to assign a new process for all incoming TCP/IP connections. The system process table is often filled completely with legitimate commands which are prevented from being executed. This attack results in a temporary failure of services. This attack could be recognised by looking for the number of active connections on a single port.

Flow characteristics of a DoS attack

Common characteristics of a DoS attack which can be detected from the traffic are:

- Sudden increase in traffic (increase in the numbers of flows).

- Changes in the traffic pattern and behaviour (e.g., server can not respond to all the requests).

- Packets incoming with particular TCP header flag combinations.

2.9.4 Overview of modern attacks

Nowadays, cyber attacks are complicated and most of them are done at the application level. Usually, prior to the attack, the attacker performs some foot-printing, which might involve probing attacks, to gather information on the target. Furthermore, malware (software designed to infiltrate or damage a computer system without informing users or raising their suspicions) is used extensively. It includes the use of viruses, worms, key-loggers, and Trojan horses [6, 151].

The most popular attacks use Remote Administration Tools (RAT) (also called “Remote Access Trojan”) which consist of two main components, a client, and a server. RATs have to be fully undetectable (FUD) by antivirus programs. The user of an infected computer often has no idea that somebody else is controlling their device because nothing is visibly

happening on their machine and all the processes are running in the background. Usually, a RootKit tool is installed on the infected machine to hide this process from the list of processes in the task manager. RATs can be transported in many ways such as through spam email, malicious websites, and USB sticks. Depending on the RAT's type and function they can also spread via other malware to infect other computers and devices and control anything on them. Furthermore, RATs can be used to conduct any other type of malicious activity including making an attacker anonymous and allowing them to steal information such as passwords and user data or perform DDoS attacks as part of a botnet network. The most popular RATs are Cybergate, Darkcomet, Blackshades and jRAT. C++ RATs are generally considered to be the most utilised type because they do not require the .NET framework and support multiple operating systems such as macOS, Linux, Android, and Windows [6, 30, 36].

Phishing attacks are used to obtain data such as usernames, passwords, and credit card information [126]. A traditional phishing attack is performed using spam email spoofing or SMS messaging and involves providing URL links to redirect users to a fake website that resembles a known legitimate one. Today's DNS servers keep changing on the user's router. When that happens, all the requests the user makes on their browser get redirected. For example, when a user types the URL of a specific website, they will be redirected to a similar website which looks identical to the requested one. As such, when the user types their login or credit card information, somebody else can access and steal this information. In this context, spam is popular and can be used to send a bulk of electronic messages such as emails, instant messages, and mobile messages. Furthermore, phishing can also be carried out through botnets [65, 66].

Key-loggers can be installed on an infected machine. They can be configured to send reports of all keystrokes to an email address. Key-loggers can even extract existing information such as taking screenshots and sending them to a specific email address. For example, Key-loggers can be programmed to send a screenshot of the user's device every five seconds. Today's key-loggers are very advanced to the extent that they have a large number of fully configurable options [132, 147].

SQL Injections involve simply passing SQL queries to HTTP requests. If they are not properly formatted by the PHP code on the server side, they can present a serious a problem. Often, this is a one of the primary considerations of all web developers [35].

Botnets represent an everyday threat to Internet users. Botnets involve command and control instructions, where the bot master remotely controls bots in the compromised victim's machines (slave bots or zombies). Botnets can be implemented in a number of ways depending on the protocol used (e.g.,TCP, Telnet, IRC, PSP and others). Furthermore, a botnet

can be used for many different purposes including most of the attacks previously mentioned. The most popular botnet types are Storm, Waledac, Conficker, Zeus and BredoLab [36, 66].

In general, most of the aforementioned attacks are used to gain illegal access to other computers or networks and access users' information, accounts, financial data, or attempt to penetrate the security measures of a system or network. This includes any activity that may be used for penetration including port scans, stealth scans, or other foot-printing used to gather information [6, 151].

These different types of malicious activities require different methods of detection and investigation. Monitoring and analysis of traffic could help to recognise characteristics of such sophisticated attacks and these characteristics include:

- Sudden increase in traffic (increase in the number of flows).
- Changes in traffic patterns and behaviour.
- Repeated unusual flow characteristics.
- Unusual robotic flow behaviour.
- Unusual traffic behaviour compared to the known normal behaviour baseline of a specific network or machine.
- Unusual IP addresses and communication hosts.
- Unusual traffic entering and leaving a network.
- Probing or DDoS flow characteristics are also seen as part of sophisticated attacks.

2.10 Summary

This chapter has focused on the cyber environment and discussed the importance of situational awareness for network administrators. In addition, it has provided an overview of typical network security systems. The chapter has also detailed the types of monitoring, the technology used for detection and the security systems functionalities and limitations, in order to show where our proposed system fits within the existing security tools. Furthermore, it has described the most common types of network protocols and traffic behaviour. Finally, the chapter concluded by defining the various attack types and some of the characteristics of malicious behaviour encountered on today's networks.

Chapter 3

Sonification

3.1 Introduction

Kramer et al. stated that the sector of sonification is now in a role to leverage the new computer audio technology to fix numerous existing problems of scientific display [99]. Sonification is used to translate relationships in information or data into sounds that enable the listener to comprehend the information or data relationships. Sonification is applied in diverse fields including as audio engineering, computer science, music, and security. The motivations of sonifying information rather than a visualisation have been discussed extensively in the literature. As a conclusion, sonification may be most appropriate when the data or information intended to be displayed is complex and has changes over time and, especially, where it might include warnings for immediate action. This involves environments where the operator is unable to continuously monitor a visual display or might be busy with another task. The function of sonification was described in categories of alarms, alerts, and warnings; status, process, and monitoring messages; data exploration; and art, entertainment, sports, and exercise [76, 161].

3.1.1 Brief summary of the history of sonification

The Morse code (invented around 1837) is probably the oldest example of sonification. In the early 20th century, the Geiger counter was invented in order to create a sonic signal that indicated the levels of radiation detected by the instrument. Actually, within the research community, several sonification systems have been presented and described since the 1990s. They are different in their scope of features and limitations, as some of them were designed as laboratory equipment, meant for different specialised cases. One of earliest examples of sonification research was conducted by Pollack and Ficks [123] (in the year 1954) and was

concerned with evaluating the information transmission properties of auditory stimuli. They have evaluated two types of mappings of multidimensional data into the parameters of sound. Another early example of sonification was Speeth's [143] (in the year 1961) work to develop am auditory representaiton that allowed the differentiation of earthquakes from underground bomb blasts based upon seismic measurements [50].

In 1913, Dr. Edmund Fournier d'Albe [40] of the University of Birmingham used selenium scanning devices (photosensors) to detect black print and then convert it into audible output. In 1977, Chambers, Mathews, and Moore [33] conducted research in the field of three-dimensional audio as a way to improve scatter plots. In 1982, Sara Bly [19] advanced and developed a method of parameter mapping sonification based on multi-dimensional data sets. A decade later, the inaugural International Conference on Auditory Display (ICAD) was an important milestone for sonification in general. Since then, the amount of research related to cognition and sonification has increased, the development of models and methods of sonification and the creation of methods in various application areas.

The most recommended and popular books in the area of sonification are Kramer's 1994 volume [96] which contains expanded versions of the papers presented at the first ICAD and Hermann et al's 2011 sonification handbook [70] which provides an overview of the key research areas in the fields of sonification and auditory display. Also, Kramer et al. [99] introduced the Sonification Report: Status of the Field and Research Agenda, which provides a summary of sonification research including the then current status of the field of sonification and auditory display and a proposed research agenda. The International Conference on Auditory Display (ICAD) is still considered as one of the most popular international conferences for this research area which focuses on auditory displays and the array of technology, perception, and applications of sonification.

Over the past few years there have been a number of special editions of journals dedicated to the field of sonification. The 2005 IEEE Multimedia special issue [75] presented a series of articles dealing with topic of interactive sonification. In the same year, an issue of ACM Transactions on Applied Perception [100] republished a number of key papers formerly presented at the first ten ICAD conferences. In 2014 Organised Sound published an issue on sonification from the perspective of electroacoustic composition [134]. Interactive Sonification was the theme again for another IEEE Multimedia special issue in 2015 [42]. In 2016 the Journal on Multimodal User Interfaces [89] presented a special issue of expanded versions of papers selected from the ICAD 2015 conference dealing with recent advances in auditory display. Finally, the topic of the sonification of real time data was explored in a special issue of Displays in 2017 [160].

3.1.2 Sonification techniques

Sonification approaches are mainly based on the following techniques.

Auditory icons

Auditory icons are brief communicative sounds that can represent objects, actions, functions, or processes. Auditory icons take advantage of associating the sound and its intended meaning based on the user's prior knowledge of sound sources and causes. This technique is mostly used for alarm, warning and navigational signals [55].

Earcons

Earcons are hierarchically-organised non-verbal audio messages. They are usually used when there is no a clear association between the sound and its intended meaning. They are 'composed of motives, which are short, rhythmic sequences of pitches with variable intensity, timbre and register' [23, p. 472]. In earcon-based approaches auditory signals or signs are combined to form more complex messages, just as sentences are formed from combining spoken words. [71].

Parameter mapping sonification

The most common technique is parameter mapping sonification which comprises the mapping of facts features onto acoustic parameters of sonic activities (such as level, pitch, duration, and onset time) [76]. There are two types of parameter mapping sonification[76]:

- Discrete sonification: this involves applying a sound event for each data vector and key data features are mapped according to time.
- Continuous sonification: similar to the discrete type but acoustic parameters of a continuous sound stream are changed according to the data and key data features are mapped according to time.

Moreover, the user can interact with the mapping in two ways:

- Interactive data selection: this involves controlling in real time what datasets or data properties are to be sonified.
- Mapping interactions: this involves adjusting the parameter mappings to alter aspects of the mapping such as ranges, conditions, scaling laws, time periods, etc.

The user can then form judgements as to whether changes to the mappings or data selection have improved their understanding of the data.

Model-based sonification

In model-based sonification the sonification designer constructs a virtual model the structure of which is driven by the dataset. The user then excites the model and the sounds it emits provide information about the underlying dataset. This sonification model facilitates the auditory perception of critical structures in the data. This kind of sonification has a tendency to involve high data dimensionality and high numbers of data points [76].

Audification

Audification is typically applied to large datasets with periodic components that lie outside the human audible frequency range of 20 Hz – 20,000 Hz. In audification the data values are scaled to lie within the audible frequency range. For example, low periodicity seismic data can be audified in order to categorise seismic events [76]. Generally audification is not an interactive or real time technique and the entire dataset is usually required in advance. Therefore, it was not considered for use in this research.

3.1.3 Interactive sonification

This section concerns human interaction with sound. Not all sonification requires interaction; for example, monitoring tasks and alerting systems offer rich information to the user separately from any actions the user might be carrying out [76]. However, often the user will wish or need to control the sonification process on the basis of the data values being represented by it; this is known as interactive sonification. Hermann and Hunt [76, p. 274] defined interactive sonification as ‘the discipline of data exploration by interactively manipulating the data’s transformation into sound’ and as ‘the use of sound within a tightly closed human-computer interface where the auditory signal provides information about data under analysis, or about the interaction itself, which is useful for refining the activity’ [69, p. 20].

Human Computer Interaction (HCI) (see Fig. 3.1) is defined as ‘a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them’ [69, p. 20]. This consider studying human interaction with computer systems or computer networks to transform data or network traffic into sound for the purposes of interpreting that data or traffic.

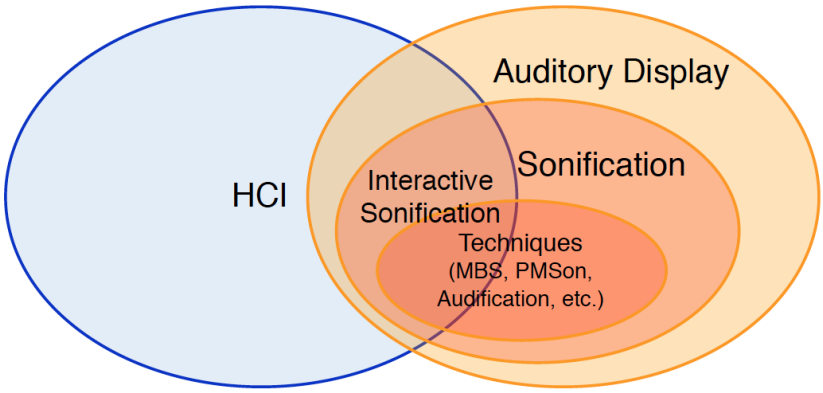

Fig. 3.1 Illustration of Interactive Sonification [76, p. 274]

Often interaction involves the operator, the operator's requirements and activities in response to recognised sounds, the means and methods of controlling a sonification system, and how the operator and the sonification system form a closed loop [76].

Hermann and Hunt [69] observed that musical instruments in particular are a good example of interactive systems in which sound plays a crucial role in coordinating the user's activities. In interactions with physical systems, they say, the feedback is natural in the sense that it reflects a coherent picture of the temporal evolution of the system. For example, this could represent the temporal state of the data under investigation to learn more about the data or for interpreting the state of the data changes with time, rather than for musical expression. In turn, the resulting understanding would affect the listener's activities, reactions, and decisions just as humans have built up experience over thousands of years of developing and performing with musical instruments [69].

3.1.4 Soundscape

The soundscape, introduced by Schafer [133], is one form of sonic organisation which, can be used in sonification. Sounds are a continuous and active property of all landscapes. For example, the sounds of vocalising and stridulating animals and the non-biological sounds of running water and rustling wind emanate from natural landscapes. The soundscape term has been used by various disciplines to represent the relationship between a landscape and the composition of its sound [121].

A significant number of the basic principles of soundscape ecology is derived from those of landscape ecology. Soundscape ecology is generated from the ecological sounds and spatial temporal patterns as they are created from landscape environment, where each sound has special ecological characteristics. For example, listening to natural sounds and human

caused sounds around us provides information about our surroundings. However, urban soundscapes are seen as containing less acoustic information than natural soundscapes [13].

Ecology is is the study of the relationship between individuals and communities within their living environment. Therefore, soundscape ecology studies the effects of the acoustic environment created by those living within it due to their responses and behavioural characteristics. The reason behind it is to identify imbalances which may have harmful or malicious effects. Soundscape composition is a reflection on the changing soundscape focusing on disappearing sounds and the increase in overall volume in present and future soundscapes [1].

3.2 Sonification for Process monitoring

Many processes and activities require monitoring. It is important to be able to monitor in real-time in order to be able to quickly deal with arising problems or unexpected events. Various visualisation and sonification techniques have been applied to monitoring situations. Real-time sonification has been studied in many areas of process monitoring applications such as industrial production processes, weather change, business process, computer program execution, network and web-server behaviour. Sonification was was shown to allow people to perceive more information than those who had visual feedback alone [72, 156].

Our auditory perception has many characteristics that make sonification suitable to process monitoring. Hildebrandt [72]suggested the following areas in which sonification might help.

- Acceleration of response to critical situations: In order to be ready to receive information regarding critical situations without delay, in progressive visualisation-based process monitoring users must focus their active attention to their observation application at all times. On the other hand, in most real life conditions there no operator will conduct process monitoring continuously as technicians and operators will mainly work on other tasks with process monitoring as a secondary activity. In such cases, auditory alerts will enable users to get immediately informed about vital situations, even without actively paying attention. Moreover, sound is processed quicker than visual signals, permitting us to reduce reaction times, which can be vital for time-critical situations which may occur throughout process execution.

- Constant awareness of process states with minimal distraction from other task: Sonification monitoring offers operators more physical freedom. However, it also allows

them to focus their visual attention on alternative tasks whilst at the same time listening to a sonification. As sounds are often processed more passively than visual information, auditory process monitoring can enable an unobtrusive perception of process states without distracting users from other tasks. As our sound perception is in a position to habituate to regular soundscapes, such a sonification will stay unobtrusive throughout ordinary operation, whereas even tiny changes in sound over time are able to immediately grab our attention in case of process deviations or undesired process behaviour.

- Anticipating critical situations: Current process control systems are based on the concept of alarms and alerts, which are transmitted either visually or using simple audio signals when a predefined threshold value has been reached. In a production scenario, this can be the case when the inventory level of the resource falls below the critical level, or when the temperature sensor of a device measures a critical temperature, indicating impending hardware failure. However, there are many drawbacks. On the one hand, if rules are set to determine the values of highly charged alerts that require strong evidence before the positive ratings are issued, potential critical situations such as device failure without alerting may occur. On the other hand, if the values are defined too liberally, meaning they risk high false positive rates, the overflow (in many cases unnecessary) alerts and alarms may result in increased user information loading. In such cases, frustrated users may decide to ignore alerts and alerts altogether. Moreover, in many scenarios, engineers cannot identify all the states and values that could lead to a predetermined critical situation. But even if all possible critical situations are covered by forecasts and warnings, operators may in most cases prefer to inform them even before the situation becomes critical, and thus the ability to anticipate, avoid and avoid the problem. Continuous awareness of countries and values through the surrounding auditory information systems can enable such expectations from critical situations.

3.2.1 Types of monitoring

Vickers [156] identified three categories of monitoring activity, namely direct, peripheral, and serendipitous-peripheral. In direct monitoring, the operator is directly engaged with the system being monitored and their attention is focused directly on the system as he takes notes of the system state. In peripheral monitoring, the operator's primary focus is elsewhere on a primary task, with their attention being switched to the monitored system either at their own discretion or at specified time periods or intervals. In serendipitous-peripheral monitoring ,

the operator's focus is on a primary task whilst information that is useful but not required is presented on a peripheral display and is monitored indirectly [156].

Whilst visualisation can be used in direct monitoring tasks (due to the fact our visual attention is focused on the display), it may not be so powerful in peripheral or serendipitous-peripheral monitoring conditions. It is in the monitoring of peripheral information that auditory displays come into their own, for the human auditory system does not need a directional fix on a sound source in order to perceive its presence [156].

3.2.2 Process monitoring review

Sonification has been applied in various disciplines, especially for purposes of real-time monitoring. In plants and factories machine maintenance technicians have been listening to the acoustic patterns machines produce for decades. They are often able to assess whether the machine is about to collapse, or a machine parts needs to be replaced soon, by listening to the frequencies and patterns of sounds produced by the machine. However, using sonification techniques, these inherent characteristics of sound can be utilised and made available to a wider range of people. On the one hand, sonification can reduce the need for the expertise necessary to analyse vibration significantly, as data can be collected and filtered according to the needs of individual user information while at the same time improving the sound resulting from our cognitive abilities. On the other hand, users who need to monitor production processes (such as engineers) often work at offices away from the factory floor, for example. In the control rooms. These rooms can contain systems that produce a large number of audio alerts that are often seen as stress. If the mechanisms of how sound can be better understood, sonification can help reduce the need for such obtrusive auditory alerts while at the same time conveying better awareness of the process situations in a more enjoyable and less stressful manner [72].

The first factory auditory process monitoring was built by Gaver, Smith and O'Shea [56]. The system was called ARKola and and provided a multi-modal sonification of a simulated soft bottling plant [156]. In this factory simulation, in an effort to avoid pauses and bottlenecks, users manually controlled the adjustments and settings of many interconnected devices or machines. Events such as liquid spills were sent to the user by appropriate sounds when they occurred. This example shows how sonification can guide operator attention in monitoring processes.

Rauterberg and Styger [128] used sonification for monitoring a production process of an assembly line of computer numeric control (CNC) robots. The original version of the system used visualisation techniques only but then was later enhanced with auditory feedback. They built a simulation of an assembly line by creating auditory icons and arranged their sonic

spectra such that perceptual masking would not occur. The CNC simulator could produce up to thirty-eight sounds at any one time. The results suggest that the additional sound feedback enhanced the operator performance and rises positively some mood aspects [156].

Tran and Mynatt [152] introduced a sonification system based on the user's own musical preferences to monitor home environments. It is a serendipitous-peripheral monitoring application. The system enabled the real-time monitoring of activities around the house. Schmandt and Vallejo [135] introduced a sonification system called ListenIN. The system was designed to have multi-domestic purposes, one of them being to detect crying babies. The system was based on classified domestic noises and sound matching.

Hermann, Drees, and Ritter [68] used sonification to present auditory weather forecasts in a regular radio programme. Bakker, van den Hoven, and Eggen [9] conducted a search for ways to leverage human auditory perception skills in interactive systems. They presented three demonstrations, called AudioResponse, EntranceSounds and RainForecasts, each of which conveyed different information. For example, The RainForecasts demonstration provided, once every half-an-hour, audible information about the short term rain forecasts for the city used in the experiment.

Donald Knuth, mentioned as early as 1949, used the interference generated by the computer's CPU to allow the execution of running programs to be followed by listening to an appropriately tuned AM radio. A computer (the Manchester Mark 1 computer) had its circuit wired to an audio channel in order to allow audio-enabled debugging [156]. Vickers and Alty [157] has proposed using music in HCI as a way to use sonification for computer program debugging.

A business process is an organised set of activities or tasks that are designed for the sake of reaching a business goal [73]. During business process monitoring, firms need to keep informed about the performances of the currently executed process and critical events that happen during the execution. Hildebrandt [72] proposed a multi-modal solution that combines sonification and visualisation. During normal operation, the system will sonify occurring events, notifications, and alerts as sound events in real time for the users. The users will be able to give the level of detail and types of events they are interested in. Simultaneously KPIs (Key Performance Indicators) are sonified by continuously updated sound streams. Several types of events will be sonified with various sounds, enabling the users to choose if they direct their immediate attention to their process monitoring application in order to take appropriate actions. The results suggest that sonification is extremely suitable to enhance visualisation in business process monitoring.

3.3 Sonification for network monitoring

Visualisation has been used as a tool for monitoring networks in order to maintain operator situational awareness to make data perceptible. Goodall [62] identified several potential issues of network and computer systems security which may benefit from information visualisation:

- detecting anomalous activity;
- discovering trends and patterns;
- correlating intrusion detection events
- computer network defence training;
- offensive information operations;
- seeing worm propagation or botnet activity;
- forensic analysis;
- understanding the makeup of malware or viruses;
- feature selection and rule generation;
- communicating the operation of security algorithms.

Each of these areas may also benefit from the application of sonification techniques.

3.3.1 Sonification for security situational awareness

Sonification is introduced to support the perception phase of situational awareness to enable a listener to recognise changes in activities and patterns to enhance comprehension and projection as part of the situational awareness process. Auditory perception has advantages as it is well suited to monitoring activities and it enables the human brain to participate in the processing required to identify certain patterns. This opens up possibilities for it as an alternative or complement to visualisation techniques. Sound may allow a network administrator to continue monitoring the network while performing other tasks [156] 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 [11]. 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 [168].

There is a continuing threat of intrusion, denial of service attacks or numerous other abuses of network resources which require the monitoring of traffic flows passing through a network [44]. The size of modern network traffic volumes makes it much harder to present real-time information visually [47]. However, there is no clear consensus yet about the patterns of cyber-attacks [82]. 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. Therefore, auditory feedback could be used to for the exploration traffic data in order to improve the situational awareness of a security system as well as its usability (efficiency, effectiveness and user satisfaction).

3.3.2 Interactive sonification for monitoring

Mechanics routinely use sound to examine the internal state of engines as whenever a user sensor interact with a physical component, a sound is made. It confirms the sensor's initial contact with the object but also informs us about its properties. The same concept, when applied to a monitoring system, would allow the user to have the best control of the sounds generated according to the user needs and targeted behaviour. The user also could change the type of information or features to be represented by sound. Moreover, the user may increase or decrease the total sound or selected sounds or even mute them. The change made by the user would affect the feedback sounds generated by the system. Sounds inform and warn the user about the current state of the system and data being sonified. Interactive sonification for computer networks can be implemented in two ways [120]:

- Interactive sonification of stored traffic data: This involves a previously saved dataset which could be explored interactively while sound is being generated.

- Interactive sonification of traffic data generated in real time: in this case the sounds are generated as the data traffic are gathered in real time. The changes in the traffic behaviour are instantly transformed into changes in sound and the user might interact to manipulate conditions or choose which data to sonify.

3.3.3 Sonification system design for network monitoring concept

When designing a sonification system for the purpose of monitoring a system or network activity to gather types of administratively useful information, the design will involve a number of conditions and requirements. The sound should be non-fatiguing to enable it to be listened to for a long period, changes in status have to be easily grasped and accidental events have to be immediately noticed [91]. Despite the close relationship between communities of sonification, there are no agreed design guideline for sonification or listening sound concepts [64].

Designing a sonification system for network monitoring requires knowledge of computer networks in multiple areas and sound design alternatives as well as a clear understanding of how users will use the system. A user-friendly sonification system is required to allow users to specify data mappings and to interact with the sound generated in order to observe, orient, decide and act. One approach could be to use a soundscape to transform the network environment to a (familiar) acoustic scene such as using the sounds of different birds and natural events in a forest.

A good sonification system design using known synthesised or recorded sounds would allow the user to make links between the meaning of the sounds and the events happening within the network environment. This requires knowledge of how best to convey the features of network events using recorded sounds. For example, a normal forest behaving normally on a normal day would have sounds with normal birds or animals, perhaps with a very light breeze. These events can be used to describe the normal state of a computer network environment. However, human-made sounds or weather such as rain, heavy rain and thunder or sounds from fire and dangerous animals could reflect anomalous events happening within the environment. Indeed, Ballora et al. commented that efficient sonification of network events or behaviour could be expected to sound different during intrusion attempts [11]. Recorded sounds provide high potential for using a natural soundscape to create an interactive acoustic environment.

3.3.4 Overview of existing sonification systems for network monitoring

There have been several attempts to investigate the use of sonification in network monitoring applications and sonification has the potential to allow a higher level of SA. However, there is no clear idea about the pattern of cyber-attacks [82]. The assumption is, therefore, such that in sonification, the behaviours and the rhythms of each type of attack should sound different or at least that sonification should provide an indication of some features of the attack.

Vickers et al. [159] 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 SOCS, the self-organised criticality sonification system, to sonify the log returns of packet sizes at regular user-specified intervals. The extracted log returns provided information on the behaviour changes in the network. Knowledge of this information could be used to detect unwanted behaviour. This system has potential to support both network traffic measurements and intrusion detection tools.

Worrall [167] described the NetSon project from its exploratory stage to the real-time sonification of network metadata. This project used the information extracted from data volumes by employing sampling techniques to extract a small group of data packets using the sFlow tool [80, 119]. This method provides information about the network flow rate by carrying out a sonification of sFlow packets data of the traffic from printers, servers and load balancing traffic. Furthermore, NetSon uses a parameter mapping sonification based on a melodic pitch structure. 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. [108] used sonification to help "cyber defenders" to detect evidence of cyber attacks by simply using data collected by Wireshark. The data was used offline and the source and destination IP addresses were sonified using pairs of sequential musical notes separated by 100 ms, while the packet size was used to control the loudness of the sound generated. An experiment revealed no improvement in the operator performance when using sonification. However, it could be argued that sonification should be tailored such that traffic with specific signatures should sound different from other normal packets, or that sounds should be generated only for malicious signatures. This may enhance the performance and reduce the stress of the operator.

Wolf and Fiebrink [165] developed an offline sonification tool called SonNet, which consisted of a code interface for sonifying computer network data. The prime motivation behind SonNet was to lower the practical barriers between 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. Furthermore, 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.

Rutz et al. [130] introduced the SysSon platform for developing sonification applications for different types of users from domain scientists to sonification researchers, composers and sound artists. This system can be used as an engine to run real-time sounds based on available data. Rutz et al. also used SysSon based on climate data. SysSon is capable of addressing network traffic data based on metadata, features or information that can be derived from network traffic.

InteNtion (Interactive Network Sonification) [60] is a project targeted at mapping network activity to musical aesthetics. Data collected from the analysis conducted by the SharpPCap library (part of WinCap to C# environment) [111] is converted into MIDI messages and then sent to dedicated synthesisers to generate mixed sounds. The whole process results in an interactive soundscape. Furthermore, the system uses IP Internet protocols including TCP/UDP segments, and very low-level packet information such as the packet size, source and destination IP addresses and the type of service. However, the work and mapping carried out at this stage is still considered to be experimental. 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 [12] 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. [11] conducted another sonification experiment with a computer network based on socket connections using information such as the date and time of exchanges and the sender's and receiver's IP addresses and port numbers. Ballora et al. [10] also described the use of sonification in the detection of anomalous events. However, there is still a need to reflect the overall SA required by network administrators. Sonification should enable the listener to differentiate between normal and anomalous network behaviour. As such, cognitive processes would allow an administrator to recognise different sound patterns from the network behaviour and translate them into an understanding of what is actually happening in the network.

Garcia-Ruiz [54] proposed a multimodal human-computer interface to analyse malicious activities during forensic analysis of IDS log files. Two prototype sonification systems were built that map attacks already identified in the IDS log files to sound. Garcia-Ruiz [53] developed the multimodal technology to be used for teaching students to analyse the network traffic data in logs and to identify patterns of network attacks.

Gopinath [63] proposed to study the effectiveness of sonification in network intrusion detection systems to support Snort. Jlisten is an open source tool to sonify Java programs.

Jlisten was used to sonify several events in Snort for the purpose of investigating the usage of sonification in these systems.

Kimoto and Ohno [91] introduced the Stetho network sonification system, which was aimed at system administrators. Stetho is a C language program linked with the TiMidity++ software to generate MIDI sounds. NetSound was built on top of Stetho as a tool for end users. Stetho uses network traffic information to generate sounds, which provides the network administrator with useful information about the traffic. Furthermore, Stetho reads the tcpdump commands, then uses them in regular expressions to generate corresponding MIDI events. Stetho also 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 [32] developed a tool for the measurement of round trip time by 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. Furthermore, they designed a stream-based method for the direct display of critical qualities to the ear by continuously driving a bidirectional connection to create sound waves. They also changed the network path to an acoustic medium, which their probe sets into vibration. Temporal levels of musical foreground, middle ground and background could thus be heard in the melodies generated from correspondence data.

Gilfix and Crouch [59] introduced a sonification network monitoring system called 'Peep', which plays different natural sounds, where each type of sound represents a specific network event. Peep easily allows the detection of common network issues such as high load, excessive traffic, and email spam, by comparing the sounds being played with those of normal network behaviour.

Hildebrandt and Rinderle-Ma [74] suggested combining sonification with existing visualisation techniques in order to tackle the disadvantages of current monitoring and analysis tools of security data. They mentioned the fact that the increasing amount of traffic poses a challenge for network and server administrators. Furthermore, they concluded that sonification is more suitable for data that changes over time because of its inherent time based nature and its link to human auditory cognition, while visualisation is more suitable for spatial information such as network topology maps, and data that is not time based.

Table 3.1 provides a classification of existing network sonification systems in terms of the sonification mode and purpose; in addition to the targeted features and information and the detection mode used. Online mode means that it is performed in real-time with

appropriate processing delays or intervals, while offline mode indicates that traffic or datasets are collected and saved using any available applications first, and then sonified later.

Table 3.1 Classification of Existing Network Sonification Systems in Terms of Sonification Mode, Purpose, Target and Detection Mode.

Author	Year	Name	Sonification Mode	Sonification Purpose	Sonification Targeted	Detection Mode
Vickers [159]	2017	SOCS	Offline	Network traffic monitoring for intrusion detection	Log returns of a specific time dependent concerning number of packets and bytes sent and received	Anomaly
Worrall [167]	2015	NetSon	Online	Network traffic monitoring for traffic measurements and IP addresses identification	Network metadata extracted from data volumes and source and destination IP addresses	Anomaly
Mancuso [108]	2015		Offline	Increase operator capabilities such as increase performance and decrease stress in traffic analysis process for intrusion detection	Source and destination IP addresses with packet size	Signature
Wolf [165]	2013	SonNet	Online	Sonification of network data to create music	Various levels from packet level information to network state information	
Giot [60]	2012	InteNtion	Offline	Network traffic monitoring for intrusion detection	TTL, the packet size, source and destination IP addresses and type of service..etc	Anomaly
Ballora [11]	2011		Online	Network traffic monitoring for intrusion detection	Socket connections exchange information such as date, time, IP addresses and ports numbers	Anomaly
Ballora [12]	2010		Online	Network traffic monitoring for intrusion detection	IP addresses and return codes	Signature and Anomaly
Garcia-Ruiz [54]	2007	Multimodal Technology	Offline	Analysing malicious attacks through forensic sonification to the network logs	Logs generated by Network IDS	Signature
Garcia-Ruiz [53]	2008	Multimodal Technology	Offline	Sonification to support teaching of network intrusion detection	Logs generated by Network IDS	Signature
Gopinath [63]	2004	JListen	Online	Improving the accuracy of intrusion detection systems	Sonifying the *events* generated by *Snort* IDS through event sound mappings	

Continued on next page

Table 3.1 – *Continued from previous page*

Author	Year	Name	Sonification Mode	Sonification Purpose	Sonification Target	Detection Mode
Kimoto [91]	2002	Stetho	Offline	Network traffic monitoring for intrusion detection	Tcpdump commands used in regular expression	Signature and Anomaly
Chaf [32]	2001		Online	Measurement of quality of service of a network path	”ping” utility events correspondence data using a stream-based method	Anomaly
Gilfix [59]	2000	Peep (The Network Auralizer)	Online	Monitoring a specific types of network events	Peep protocol using *events* and *states* to order Peep servers to play classified natural sounds	Anomaly

3.4 Discussion

These different sonification approaches and scenarios and their different levels of data extraction have not been tested for monitoring intrusion detection. Most of the systems are based on reading saved information files or log files which are generated by other software and security systems. Therefore, the operator does not participate in the development of the primary feature-extraction process from the raw traffic and has limited traffic information to work with.

Furthermore, these different approaches with their various data granularities, sonification techniques and integrations with visualisation have had very limited testing with users. Therefore, it remains unclear which approach is best suitable for which kind of scenario and how effective the different systems are in achieving their goal to support users in their tasks.

There are many ideas that show a strong potential for using sonification for network monitoring. However, no one system has yet been designed to deal with raw traffic packets and act as a monitoring tool that can be compared to the benefits obtained from the existing visualisation tools. There are no existing sonification systems using information at the protocol level to target traffic status based on each packet’s specific function. Therefore, it is unclear which sonification approach is best suited for real-time monitoring and how effective sonification systems are in reaching their goal of supporting users in monitoring and analysis tasks. Nevertheless, sonification has a strong potential for real-time monitoring of computer networks in order to raise the overall cyber security situational awareness. Moreover, sonification based on network traffic protocol information and mechanisms is vital in reflecting the mapping approach based on human understanding.

The state of research would be more advanced if users were involved in monitoring tasks using real-time traffic. Moreover, developing a sonification system has the potential to detect and investigate the increasing modern threats and as such, evaluation studies are necessary to prove the potential of sonification in computer network monitoring. Therefore, the question that this research addresses is 'how can sonification be used in maintenance of real-time situational awareness to provide the protocol flow granularity required to understand the network environment behaviour?'.

3.5 Summary

This chapter has introduced sonification and explained its approaches and techniques. It has also presented a brief discussion of interactive sonification, soundscape, and soundscape ecology used in the development of the SoNSTAR system. Furthermore, the chapter has also described the areas where sonification can replace or support visualisation as a tool for monitoring. In addition, the chapter has explained in which phase sonification could support situational awareness and presented a concept for a sonification system design for network monitoring. Finally, a literature review has been conducted to provide a summary of existing related work on network sonification.

Chapter 4

SoNSTAR

4.1 Introduction

The sonification of high speed computer networks demands both high throughput and flexibility to handle and recognise new threats. Such systems should deal with raw network traffic (packets) and try to establish a sonification model that can enable a human operator to recognise the difference between normal traffic and anomalous activity. 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 is designed to 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) through the use of diagnosis to support the recognition phase in the situational awareness process.

This chapter introduces SoNSTAR and looks at the user specifications for the monitoring tool suite and its primary design goal of sonifying TCP traffic which is a priority as it accounts for most network traffic (see section 4.3 below). Some of the specifications are set due to sonification capabilities to support real-time monitoring. This chapter begins with the requirements for SoNSTAR and then sets out its design. The chapter concludes with a validation of SoNSTAR's soundscape approach to sonification.

4.2 SoNSTAR and Network Traffic Sonification

A traffic flow is defined as a series of packets belonging to a single connection between a source host and a destination host [25]. Each TCP packet contains header information including source IP, source port, destination IP, destination port, and protocol. Thus, a single flow can be identified within a certain time period by its source and destination IP addresses, its source and destination ports and its protocol layer (such as TCP, UDP and ICMP). As part of our technical solution, we have created new flow type called IP flow which is identified within a certain time period by its source and destination IP addresses and protocol. SoNSTAR uses these two flow types (traffic flow and IP flow) in its sonification approach. A host connection is the set of all traffic flows or IP flows passing through a specific host connection which could be a single device or node or switch. Network traffic is the set of all flows passing through the network.

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 a flow's features while a set of events represents a flow's behaviour which represents the state of the network traffic.

In the TCP protocol, the header contains nine control flags, six of which (FIN, SYN, RST, PSH, ACK and URG) are used by SoNSTAR. The values 1 and 0 denote whether a flag is set or unset respectively, 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 network traffic by playing sounds that represent the flow behaviours.

Thus, SoNSTAR makes information about traffic perceptible, in turn 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 behaviour in a human understandable form.

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 using pre-recorded sounds. The network administrator can then interact directly with the network environment. The method includes application of interactive sonification to a computer network using the concept of a soundscape. 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 in a forest would be able to infer information about what is happening in the 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 [98]. 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 birds tweeting or animal sounds are easier to describe than artificially synthesized tones which may rely on specific terminology such as frequency and timbre [166]. The sounds provide us with immediate awareness of the types of events that are happening. Modern cognitive science believes that to be able to study sound in this way, the listener must have some inner understanding of how the features of physical events are reflected in the sounds they make [97].

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 rainy day will produce sounds of rain, wind and thunder, perhaps with a fire on the forest. These events can be used to describe the normal state of a computer network environment. On the other hand, 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 sonification is 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.

4.3 SoNSTAR and Network Traffic Monitoring

Commonly, administrators try to look directly at network traffic to understand it using tools such as Wireshark [164]. 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 packets carry control flags to allow the data to be received in sequence and

to protect it from loss. In TCP, 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 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 [138]. 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 represents more than 85% of packets entering and leaving a system or computer network [138, 142]; therefore, TCP traffic is considered a priority for sonification purposes.

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 [14].

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 collects traffic information periodically.

The TCP control packets SYN, SYN-ACK, ACK, FIN and RST provide most of the information about network traffic state. 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.

4.4 SoNSTAR Requirements and Design

Computer network defence needs traffic analysts to identify both known and novel malicious activities and attacks in huge volumes of network traffic. Visualisation tools would potentially support the detection of malicious traffic patterns of the network, but few traffic analysts so far are leveraging sonification techniques in their current security practice. SoNSTAR is designed to suit those traffic analysts' needs.

In many systems, changes in performance could be used to indicate the vulnerability or robustness of a computer network [39]. 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 learn the normal behaviour of the target 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.

4.4.1 Monitoring requirements of the tool

Some special requirements, especially when using sounds to refer to changes in traffic behaviour, are necessary with such monitoring tools. 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 forensics analysis, visualisation and sonification tools for monitoring are given in the literature [8, 16, 28, 107, 113] 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 to 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.

It is also important that the system can read traffic datasets in common formats.

4.4.2 Background and principle

Port scans can be a sign of many attacks. A port scan is used to collect information about the scanned ports. For example, consider a SYN scan in which a TCP packet with SYN flag set is sent to the destination host. If the port at the destination is open, it will respond with either a SYN-ACK packet (if it is accepting the connection) or a RST packet (if it denies the connection). If the port is closed, it will respond with an ICMP packet indicating the port is unreachable. Usually, if botnet zombies are installed stealthily on a system, their communications may generate a port scan.

TCP communication is controlled by flag state. Therefore, a sniffer module is required to collect the network traffic. This traffic is then filtered to identify the TCP packets. The system extracts each type of flag state for each packet and aggregates the number of each packet type for each flow count in each time window period. Then features are extracted for each flow from each flow state aggregate. The number of flows generated in each time window varies, depending on the number of connections made and the duration of the time window [48, 95, 142]..

The number of flows generated in each time period can be very large and difficult to represent in real time using visualisation techniques. Sonification has the potential to offer real-time traffic representation. The sounds produced are based on events derived from the flow features and can potentially enable the operator to recognise changes in the traffic behaviours. An initial set of flow features was chosen to create events, and these events are represented by different sounds in a soundscape. As traffic passes through the network the soundscape changes in response.

Ohsita et al [115] classified TCP flows into five groups as shown in Table 4.1. This classification assists with the design of events based on traffic features. Table 4.2 shows the result of their experiment of traffic classification where they used the traffic generated by the internet in their campus network for five days period [115]. The experiment supports our intention to monitor TCP/IP traffic.

Soniya and Wiscy [142] used packet counts and neural networks for detecting SYN scans. Detection relied on the three-way handshake mechanism for establishing and termination of the connection. The experiment was based on detecting a TCP SYN port scan on a single machine. Information in the TCP flags was used to define the behaviour of a legitimate connection. Any deviation from the defined normal behaviour (which was used to train to a neural network) is used to detect the scan attack. The experiment tracked various flag counts in both directions as follows:

Table 4.1 TCP flows classification [115]

Group	Description of flows
Group N	Flows that completed the 3-way handshake and were closed normally by a FIN or RST packet at the end of the connection.
Group Rs	Flows terminated by a RST packet before a SYN/ACK packet was received from the destination host. These flows were terminated this way because the destination host was not available for the service specified in the SYN request.
Group Ra	Flows terminated by a RST packet before an ACK packet for the SYN/ACK packet was received. These flows were terminated this way because the SYN/ACK packets were sent to a host that was not on the Internet.
Group Ts	Flows containing only SYN packets. These flows are not terminated explicitly (i.e., by RST/FIN packets) but by a timeout. There were three reasons that flows could be classified into this group. One was that the destination node did not respond with a SYN packet. A second was that the source address of the SYN packet was spoofed and the destination sent the SYN/ACK packet to the spoofed address. The third was that all of the SYN/ACK packets were discarded by the network (e.g., because of network congestion).
Group Ta	Flows containing only SYN and its SYN/ACK packets. Like Group Ts flows, these flows were terminated by a timeout. In this case, however, it was because all the ACK packets were dropped.

Table 4.2 TCP classification of flows [115]

Group	Number of flows	Percentage
Group N	18,147,469	85.1
Group Rs	622,976	2.9
Group Ra	75,432	0.3
Group Ts	2,435,228	11.4
Group Ta	2,009	0.0

- C1 is the count of incoming SYN packets.
- C2 is the count of outgoing SYN-ACK packets.
- C3 is the count of outgoing RST packets.
- C4 is the count of outgoing SYN packets.
- C5 is the count of incoming SYN-ACK packets.
- C6 is the count of outgoing FIN packets.
- C7 is the count of incoming FIN packets.

SoNSTAR system has been developed to represent the behaviour of flows in network traffic at the end of a specific time window. The system collects the packet counts for each flag type for each flow during the time window. One of the contributions is the creation of a new type of traffic flow, called "IP flow" to represent the network connections based on the packet counts and the number of flows. An IP flow is identified by collecting the packet counts based on on the flag types of all the traffic between two hosts.

4.4.3 Development process and design rationale

As the research concerns the use of sonification to support the situational awareness of network behaviour and to support detection of malicious behaviour it was important to choose which protocol to start the study. The TCP/IP protocol was selected because it represents the majority of incoming and outgoing traffic on a typical network. Therefore, a real-time traffic sniffer is needed that can filter out everything but TCP/IP packets and pass them onto the sonification system. Since the aim was to explore the behaviour of the network as well as to identify the events going on within it, it was necessary first to determine what data or information to use to express the types as well as the movement and volumes of network traffic. This was one of the biggest problems encountered in determining which features to use. A number of options was considered but TCP control packet flags were the primary characteristics for representing the state of a network based on the mechanisms of the TCP/IP protocol. In particular, the status of the flags in packet headers (such as SYN, SYN-ACK, ACK, FIN, RST and PSH) were investigated to discover how they change and relate to the network traffic state.

The next stage involved the creation of the features extractor and also the features combiner for creating new combinations of existing features. At this point, the challenge was how to extract features from the TCP control packets and arrange them as traffic flows. The

solution to this was to aggregate packet status information to organise packets into multiple flows. In this way, it was possible to arrange each packet count type according to their flow based on IP addresses and ports. Now that packets were organised into flows the features of each flow could then be extracted. Events were constructed to represent the three-way handshake mechanism. However, the number of flows was very large and it was not possible with this approach to target vertical or horizontal behaviour, but some basic behaviours (such as handshakes) could be seen.

This next stage involved sonification. Despite the use of recorded sounds to represent events, the resulting sounds were many and overlapping (e.g., multiple rain sounds occurring at the same time), and a solution to the problem of representing a large number of flows while reducing the number of sounds yet allowing events to be easily distinguished needed to be found. During initial development testing, SoNSTAR was used to collect the incoming and outgoing packet counts of one device without relying on the port numbers. The result showed a reduction in the number of normal flows in such a way that the state of the traffic could be recognised. After several confirmation experiments, this new type of flow was adopted, and it was named IP flow.

The larger part of the design problem now solved, the task of finding and defining appropriate events for representing the behaviour of the flows within the network was addressed. One of the first discoveries was that the number of normal flows and the number of IP flows were also affected by changes in network behaviour. Difficulties were encountered in sonifying the events in a way that allows the user to easily recognise and learn sounds. This problem was tackled by taking all the flows with repeated occurrences of a single event and sonifying only one of them during each time period.

From the above, the complete SoNSTAR architecture was thus designed around five components: the sniffer, the filter, the features extractor, the features combiner and the sonification engine. These are discussed in detail in the following sections.

4.4.4 SoNSTAR user manual

This interactive sonification system consists of two main sections. Part 1: **Main program** written in Python. Part 2: **Max/MSP patch**, the sound control panel and the sound component output using the program Max/MSP. The primary system default configurations are presented in Table 4.7. In order to understand the configurations, The user should learn the features used in SoNSTAR. The complete features are presented in Tables 4.3, 4.4, and 4.5. This brief guide explains how to use SoNSTAR for monitoring.

- **Launching SoNSTAR:** the system can executed by launching the main Python script and by opening the Max/MSP patcher in Max/MSP.
- **How SoNSTAR works:** when the main program is launched, the system will request from the user the chosen time window period. Then the system will display the Capture Option Interface list. The user needs to select which interface the system should capture network traffic from.

  Immediately, the system will start sniffing the traffic and check traffic events and pass event messages to the Max/MSP patch. The sound will be played according to the traffic-to-sound mappings defined in the system configuration file.
- **Changing the configurations:** The user can change the conditions and the values of the thresholds of the events in the events sonification section in the main script and relaunch with the new changes. If the user would like to add a new event, they should write the event conditions and assign an unused port to the event in order to connect with the Max/MSP patch. In the Max/MSP patch, the user needs to create a new port receiver with the same port number assigned to the new event.
- **Sound control panel options:** The user can change in real-time the sound volume of each event through the control panel by adjusting a slider control. Also, the user can change the sound of any event by uploading a new sound file to that event sound buffer or use an existing sound name from the list of existing event sound buffers.
- **The sound meaning:** The meanings of the default event-sound mappings are presented in Table 4.6. The training tables at Appendix 5 and 6 provides examples of events and how they could be related and understood.
- **Terminating the system:** The system can be terminated by closing the Python script and turning off the sound generation in Max/MSP.

The system is available in full at https://github.com/nuson/SoNSTAR.

4.4.5 Design solution

This section considers a design that fulfils 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 4.1. The system is implemented in Python using the pcapy and dpkt libraries and Max/MSP. The Python engine captures and

processes the packet information and passes data to a Max/MSP patch which generates the audio (see the project repository [41] for SoNSTAR source code file).

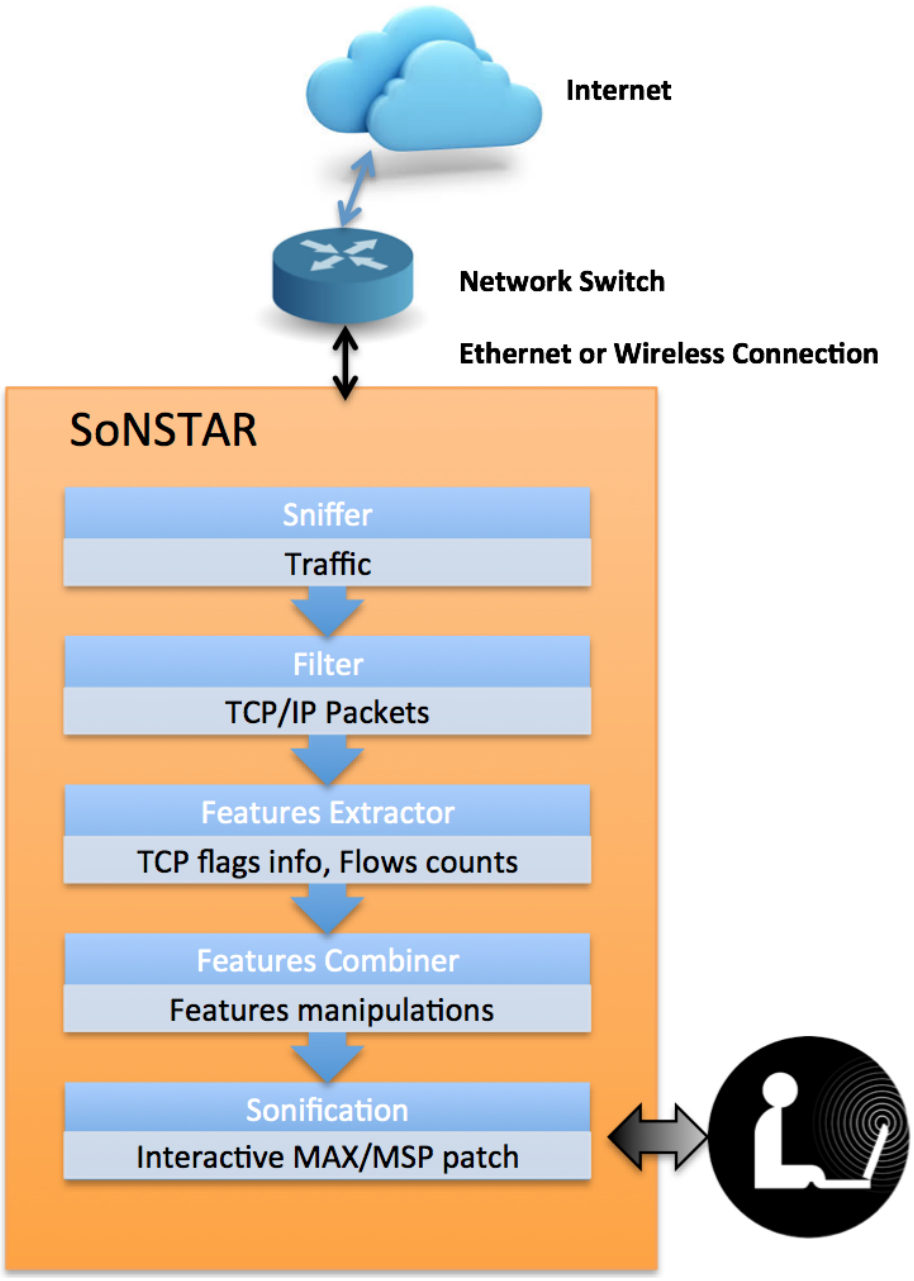

Fig. 4.1 **SoNSTAR Architecture.** The major components of the system.

SoNSTAR uses a time window period to arrange and control the timing of the operation of each process within the system (see Fig 4.2). 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*.

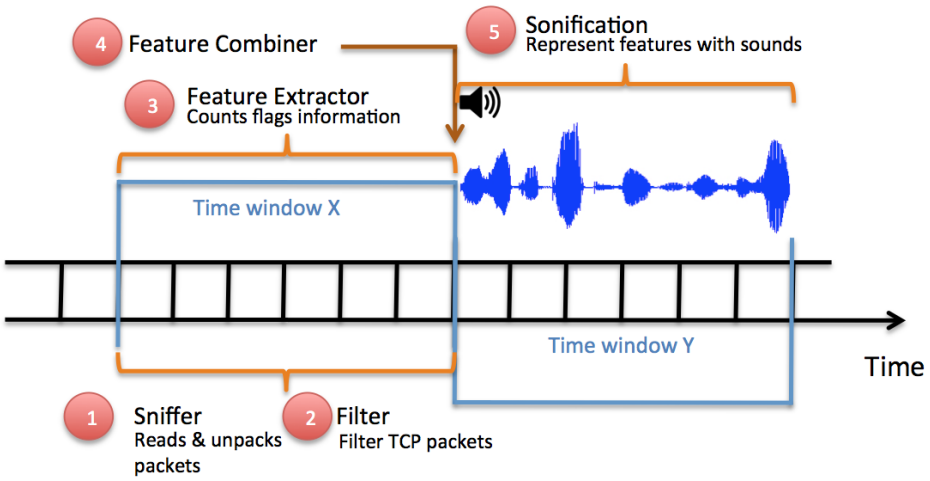

Fig. 4.2 **Time window processes.** SoNSTAR aggregates flow data across time windows. This figure shows the process timing and sequencing across two time windows, *X* and *Y*.

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

Sniffer

The main input to the system is the raw traffic packets passing (incoming and outgoing) through the network. 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

Algorithm 1 SoNSTAR's main algorithm

```
Set Time-window 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 Time-window 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
```

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.

Traffic Flows identified by `src addr`, `src port`, `dst addr` and `dst port` are followed by columns with flag status counts. SoNSTAR retains the information in an array carrying all feature information extracted from the packets in each time window for each Traffic flow as per Table 4.3.

Table 4.3 Feature information array: Traffic flow

Element	Label	Description
1	**Flow Counter**	This represents the number of flows in each time window.
2	**Address 1**	This is one of the IP addresses of the flow which changes to be the source or destination according to the side sending or receiving the packet.
3	**Address 2**	This is one of the IP addresses of the flow which changes to be the source or destination according to the side sending or receiving the packet.
4	**Port 1**	This is one of the ports of the flow which changes to be the source or destination according to the side sending or receiving the packet.
5	**Port 2**	This is one of the ports of the flow which changes to be the source or destination according to the side sending or receiving the packet.
6	**FIN Out**	Counts of outgoing FINs packets, which represent the total counts of outgoing packets which hold FIN status set to 1 and the status of other flags is set to 0.
7	**FIN In**	Counts of incoming FINs packets., which represent the total counts of incoming packets which hold FIN status set to 1 and the status of other flags is set to 0.
8	**SYN Out**	Counts of outgoing SYN packets, which represent the total counts of outgoing packets which hold SYN status set to 1 and the status of other flags is set to 0.
9	**SYN In**	Counts of incoming SYN packets, which represent the total counts of incoming packets which hold SYN status set to 1 and the status of other flags is set to 0.

Continued on next page

Table 4.3 – *Continued from previous page*

Element	Label	Description
10	**SYN ACK Out**	Counts of outgoing SYN-ACKs packets, which represent the total counts of outgoing packets which hold SYN and ACK status set to 1 and the status of other flags is set to 0.
11	**SYN ACK In**	Counts of incoming SYN-ACKs packets, which represent the total counts of incoming packets which hold SYN and ACK status set to 1 and the status of other flags is set to 0.
12	**RST Out**	Counts of outgoing RSTs packets, which represent the total counts of outgoing packets which hold RST status set to 1 and the status of other flags is set to 0.
13	**RST In**	Counts of incoming RSTs packets, which represent the total counts of incoming packets which hold RST status set to 1 and the status of other flags is set to 0.
14	**ACK Out**	Counts of outgoing ACKs packets, which represent the total counts of outgoing packets which hold ACK status set to 1 and the status of other flags is set to 0.
15	**ACK In**	Counts of incoming ACKs packets, which represent the total counts of incoming packets which hold ACK status set to 1 and the status of other flags is set to 0.
16	**PSH Out**	Counts of outgoing PSH packets, which represent the total counts of outgoing packets which hold PSH status set to 1 and the status of other flags is set to 0.
17	**PSH In**	Counts of incoming PSH packets, which represent the total counts of incoming packets which hold PSH status set to 1 and the status of other flags is set to 0.
18	**PSH ACK Out**	Counts of outgoing PSH-ACK packets, which represent the total counts of outgoing packets which hold PSH status set to 1 and ACK status set to 1 and the status of other flags is set to 0.

Continued on next page

Table 4.3 – *Continued from previous page*

Element	Label	Description
19	**PSH ACK In**	Counts of incoming PSH-ACK packets, which represent the total counts of incoming packets which hold PSH status set to 1 and ACK status set to 1 and the status of other flags is set to 0.
20	**URG Out**	Counts of outgoing URG packets, which represent the total counts of outgoing packets which hold URG status set to 1 and the status of other flags is set to 0.
21	**URG In**	Counts of incoming URG packets, which represent the total counts of incoming packets which hold URG status set to 1 and the status of other flags is set to 0.

IP flows are identified by `src addr` and `dst addr` and are followed by columns of flag status counts. SoNSTAR retains the information in an array carrying all feature information extracted from the packets in each time window for each IP flow as per Table 4.4.

Table 4.4 Feature information array: IP-flow

Element	Label	Description
1	**IP Flow Counter**	This represents the number of IP flows in each time window.
2	**Address 1**	This is one of the IP addresses of the flow which changes to be the source or destination according to the side sending or receiving the packet.
3	**Address 2**	This is one of the IP addresses of the flow which changes to be the source or destination according to the side sending or receiving the packet.
4	**FIN Out IPs**	Counts of outgoing FINs packets between two IP addresses for whole ports, which represent the total counts of outgoing packets which hold FIN status set to 1 and the status of other flags is set to 0.

Continued on next page

Table 4.4 – *Continued from previous page*

Element	Label	Description
5	**FIN In IPs**	Counts of incoming FINs packets between two IP addresses for whole ports, which represent the total counts of incoming packets holds FIN status set to 1 and the status of other flags is set to 0.
6	**SYN Out IPs**	Counts of outgoing SYN packets between two IP addresses for whole ports, which represent the total counts of outgoing packets which hold SYN status set to 1 and the status of other flags is set to 0.
7	**SYN In IPs**	Counts of incoming SYN packets between two IP addresses for whole ports, which represent the total counts of incoming packets which hold SYN status set to 1 and the status of other flags is set to 0.
8	**SYN ACK Out IPs**	Counts of outgoing SYN-ACKs packets between two IP addresses for whole ports, which represent the total counts of outgoing packets which hold SYN and ACK status set to 1 and the status of other flags is set to 0.
9	**SYN ACK In IPs**	Counts of incoming SYN-ACKs packets between two IP addresses for whole ports, which represent the total counts of incoming packets which hold SYN and ACK status set to 1 and the status of other flags is set to 0.
10	**RST Out IPs**	Counts of outgoing RSTs packets between two IP addresses for whole ports, which represent the total counts of outgoing packets which hold RST status set to 1 and the status of other flags is set to 0.
11	**RST In IPs**	Counts of incoming RSTs packets between two IP addresses for whole ports, which represent the total counts of incoming packets which hold RST status set to 1 and the status of other flags is set to 0.
12	**ACK Out IPs**	Counts of outgoing ACKs packets between two IP addresses for whole ports, which represent the total counts of outgoing packets which hold ACK status set to 1 and the status of other flags is set to 0.

Continued on next page

Table 4.4 – *Continued from previous page*

Element	Label	Description
13	**ACK In IPs**	Counts of incoming ACKs packets between two IP addresses for whole ports, which represent the total counts of incoming packets which hold ACK status set to 1 and the status of other flags is set to 0.
14	**PSH Out IPs**	Counts of outgoing PSH packets between two IP addresses for whole ports, which represent the total counts of outgoing packets which hold PSH status set to 1 and the status of other flags is set to 0.
15	**PSH In IPs**	Counts of incoming PSH packets between two IP addresses for whole ports, which represent the total counts of incoming packets which hold PSH status set to 1 and the status of other flags is set to 0.
16	**PSH ACK Out IPs**	Counts of outgoing PSH-ACK packets between two IP addresses for whole ports, which represent the total counts of outgoing packets which hold PSH status set to 1 and ACK status set to 1 and the status of other flags is set to 0.
17	**PSH ACK In IPs**	Counts of incoming PSH-ACK packets between two IP addresses for whole ports, which represent the total counts of incoming packets which hold PSH status set to 1 and ACK status set to 1 and the status of other flags is set to 0.
18	**URG Out IPs**	Counts of outgoing URG packets between two IP addresses for whole ports, which represent the total counts of outgoing packets which hold URG status set to 1 and the status of other flags is set to 0.
19	**URG In IPs**	Counts of incoming URG packets between two IP addresses for whole ports, which represent the total counts of incoming packets which hold URG status set to 1 and the status of other flags is set to 0.

Continued on next page

Table 4.4 – *Continued from previous page*

Element	Label	Description
20	**URG PSH FIN Out IPs**	Counts of outgoing URG-PSH-FIN packets between two IP addresses for all ports, which represent the total counts of outgoing packets which hold URG and PSH and FIN status set to 1 and the status of other flags is set to 0.
21	**URG PSH FIN In IPs**	Counts of incoming URG-PSH-FIN packets between two IP addresses for wholeall ports, which represent the total counts of incoming packets which hold URG and PSH and FIN status set to 1 and the status of other flags is set to 0.
22	**NULL Out IPs**	Counts of outgoing NULL packets between two IP addresses for all ports, which represent the total counts of outgoing packets which hold all flags with status set to 0.
23	**NULL In IPs**	Counts of incoming NULL packets between two IP addresses for all ports, which represent the total counts of incoming packets which hold all flags with status set to 0.
24	**LAND Out IPs**	Counts of outgoing LAND packets between two IP addresses for all ports, which represent the total counts of outgoing packets which have the same source and destination IP addresses.
25	**LAND In IPs**	Counts of incoming LAND packets between two IP addresses for all ports, which represent the total counts of incoming packets which have the same source and destination IP addresses.

At the end of each time window SoNSTAR creates two log files consisting of the all the traffic- and IP-flows with their packet type counts respectively 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 4.5 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 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 [136]. 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 Table 4.4) provided by the Feature Extractor (see Table 4.5). All of these features are now available for sonification. An example showing how a feature is created is given below in Section 4.4.8.

Table 4.5 Feature Combinations.

Feature Combination	**Definition**	**Normal range**
FC 1	SYN-out-IP − SYN-ACK-in-IP	$\leqslant 4$
FC 2	SYN-in-IP − SYN-ACK-out-IP	$\leqslant 4$
FC 3	FIN-out-IP − FIN-in-IP	$\leqslant 9$
FC 4	FIN-in-IP − FIN-out-IP	$\leqslant 9$
FC 5	SYN-in-IP + SYN-out-IP − FIN-out-IP	$\geqslant$ RST-out-IP
FC 6	SYN-in-IP + SYN-out-IP − FIN-in-IP	$\geqslant$ RST-in-IP
FC 7	FIN-in-IP − FIN-out-IP − RST-out-IP	$\leqslant 9$
FC 8	FIN-out-IP − FIN-in-IP − RST-in-IP	$\leqslant 9$

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

Sonification

The final block in the system is Sonification. To make sense of the sonification we have to 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.

Fig. 4.3 illustrates the SoNSTAR Max/MSP Patch design, showing how sound messages are received to play various recorded sounds while the user can interact and control the sounds in real-time. Fig. 4.4 illustrates part of the SoNSTAR Max/MSP Patch in larger size. Samples of audio files and SoNSTAR Max/MSP Patch can be found online at the he project repository [41].

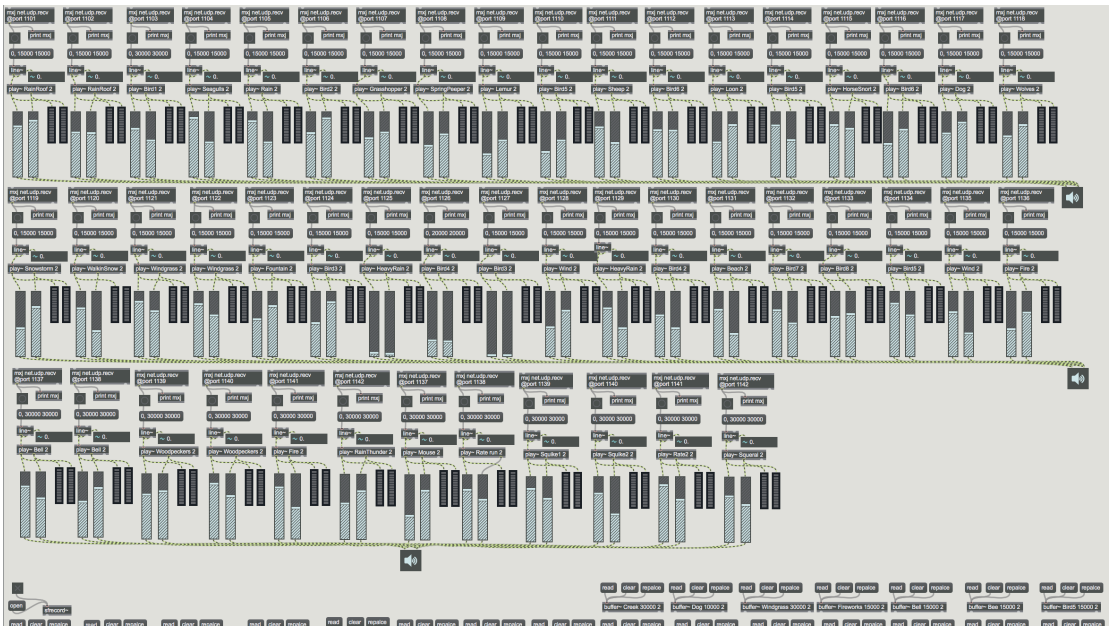

Fig. 4.3 SoNSTAR Max/MSP Patch design

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 4.5). 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 [112]. Therefore, using SoNSTAR could 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.

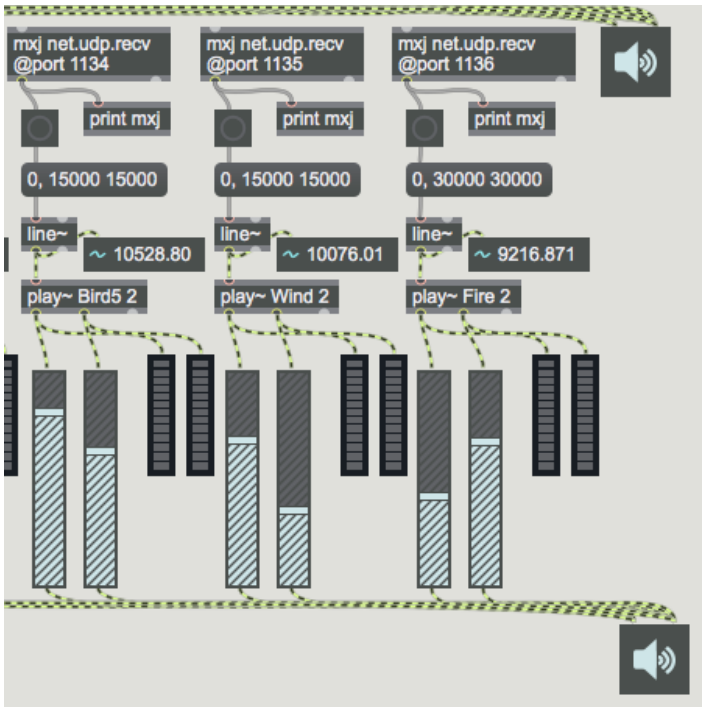

Fig. 4.4 Part of SoNSTAR Max/MSP Patch

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.

The analogy where changes in flow event, connection mechanisms and traffic behaviours would sound a similar to changes in sounds of weather or nature of animals would help administrators to recognise and comprehend behaviours easily over time.

4.4.6 SoNSTAR 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. 4.5). 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.

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.

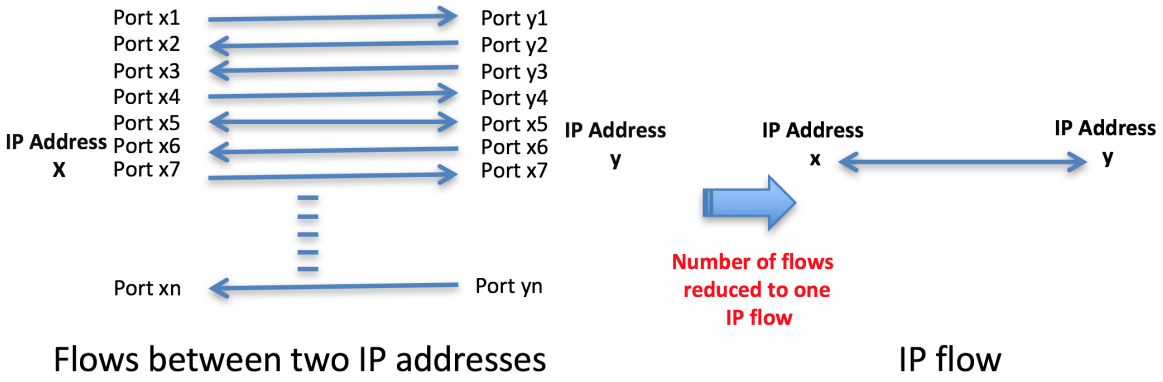

Fig. 4.5 **Conflation of multiple traffic flows to one IP flow**. Seven traffic flows between different ports on the same sending and receiving hosts are reduced to a single IP flow.

4.4.7 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. 4.6 shows an example sonification of IP flows to represent network traffic state.

(Listen to the file `DefaultSonification.mp3` of normal traffic sonification in the folder '`examples\design`' in the SoNSTAR respository [41]).

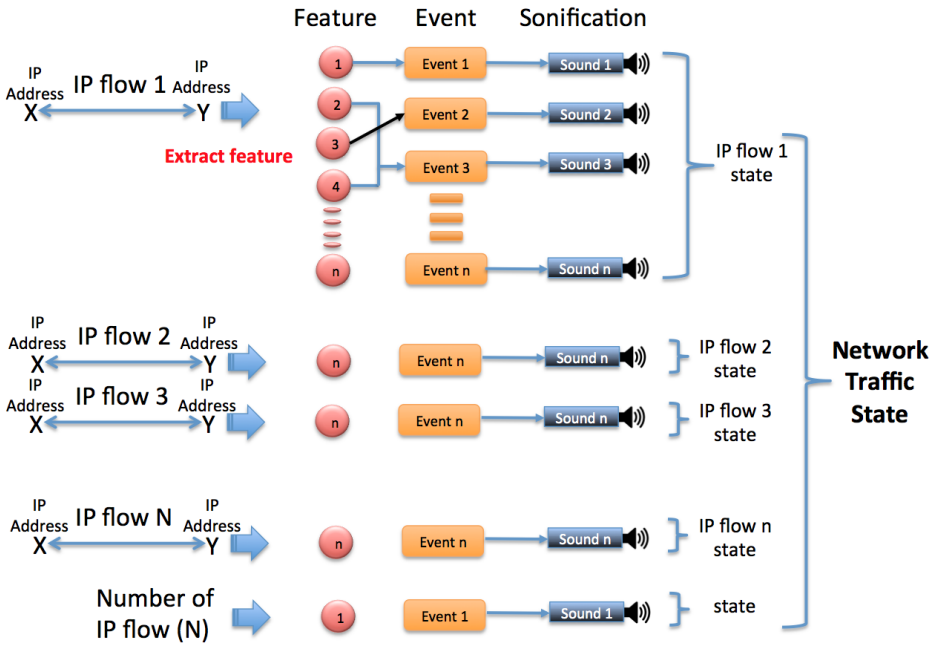

Fig. 4.6 **IP flow representation**. Illustration of multiple IP flows containing a range of different events and even combinations are mapped to different sounds resulting in a sonic representation of the overall traffic state.

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 4.7 shows an example of event representation in SoNSTAR.

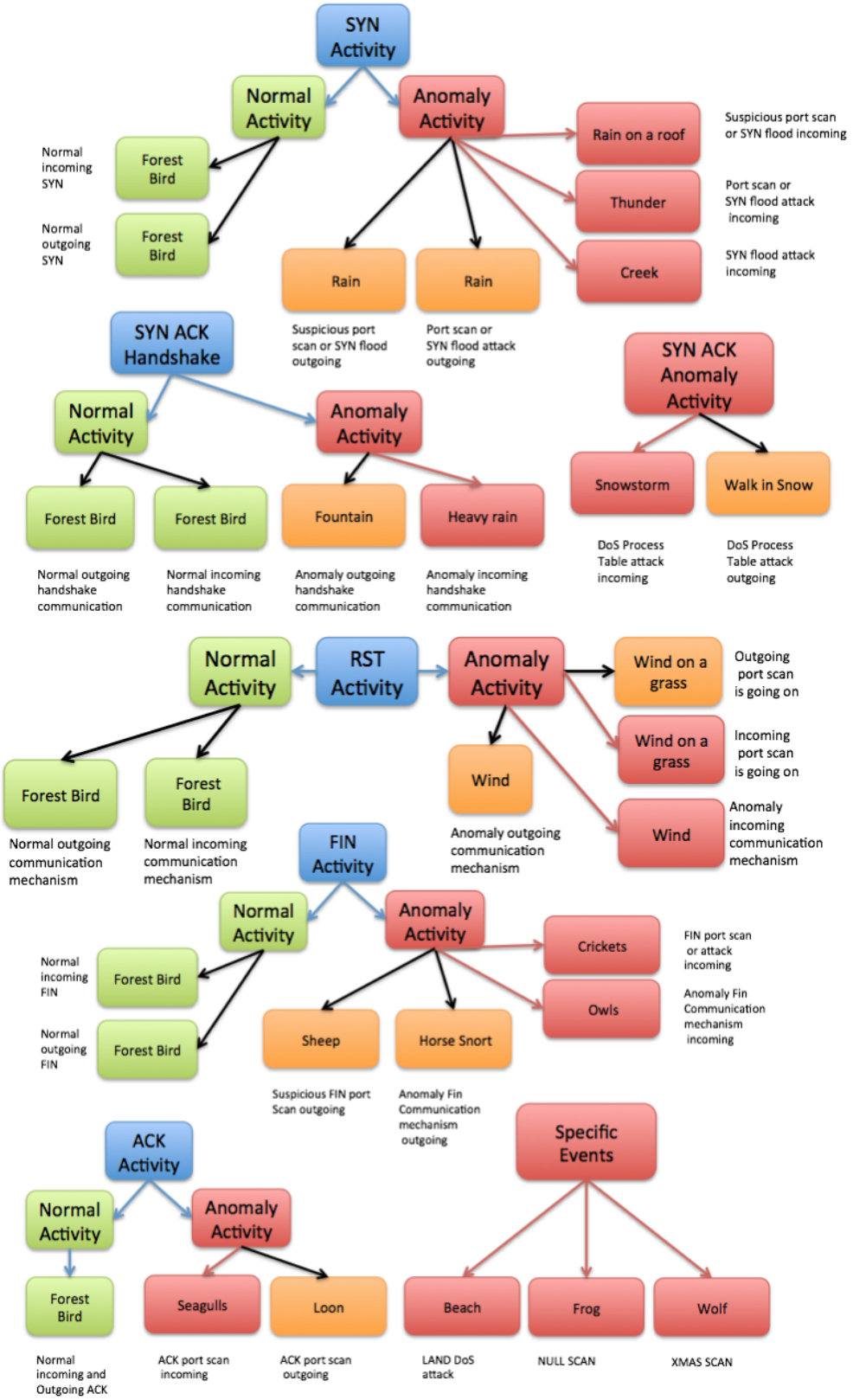

Fig. 4.7 **Event representation**. Illustration of different events (identified the main flag type) being mapped to discrete sounds in the SoNSTAR soundscape.

For a better representation, incoming and outgoing events of the same type are represented such that incoming events are given more worrying and louder sounds (more dangerous or urgent 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.

One might ask why forest sounds were chosen when most people live in towns. The forest sounds were chosen because they represent a diverse environment that provides many sounds that can be used to represent network behaviour. In addition, forest sounds can be easily distinguished and linked to their natural environment behaviour, unlike sounds in a city environment in which visual context may be needed to link a sound to a particular behaviour. For example, a person shouting in a city might be an aggressive act, or someone calling to a friend, or some other activity. The use of forest sounds also creates a separate soundscape from the one in which the user lives, thereby making it easier to distinguish SoNSTAR sounds from other sounds in the user's immediate environment. The sound of the forest used might also serve as a relaxing background that can be listened to for periods of time with less annoyance than other possible environment sounds.

Furthermore, as far as possible, sounds were chosen such that unfolding events in the traffic are understood via real world events. For example, the sound of a storm developing may start with light rainfall and then progress through heavy rain all the way to thunder and this progression is mapped to a port scan. Thus, a light scan is mapped to light rain, and the more the scan becomes a flood, so the rain sound gets heavier. This results in the creation of a special sound language between the user and the network and the events that enable them to distinguish events occurring in the network based on the feature conditions that they set to trigger the occurrence of sounds.

The user is expected to have knowledge of the forest sounds used, as well as good knowledge and understanding of the network events that the sounds represent. Table 4.6 shows the association between meanings and sounds. These sounds are understood depending on the sequence of their occurrence and one type of attack can be composed of several sounds

from other sounds in the user's immediate environment. SoNSTAR provides the operator with a way to research, develop, create and characterise new features and events. The discovered features and events may contribute to the development of IDSs and a better understanding of the behaviour of the networks.

Table 4.6 Mapping sound to meaning

No	Sound	Meaning
1	Forest birds	Each normal bird sound represents the normal condition of a normal network event which is usually heard in a forest when conditions are normal. For example, when a TCP handshake mechanism successfully completes a bird sound is played.
2	Rain	A local host is sending a number ($\geqslant 30$) of SYN packets yet none or very few of them have completed the handshake mechanism.The behaviour is like a rain of packets being sent out to the victim host.
3	Rain on roof	A local host is receiving a number ($\geqslant 10$) of SYN packets, in which none or very few of them have completed the handshake mechanism.The behaviour is like a rain of packets being received by the local host. Rain on a roof is used here to allow the user to distinguish between incoming and outgoing behaviours.
4	Heavy rain	It indicates the number of successful incoming handshake mechanisms is low compared to the received requests (uses SYN packets).
5	Fountain	It indicates the number of successful outgoing handshake mechanisms is low compared to the sent requests (uses SYN packets).
6	Thunder	A local host is receiving a large number ($\geqslant 300$) of SYN packets, where none or very few of them have completed the handshake mechanism.
7	Creek	A local host is receiving a very large number ($\geqslant 1000$) of SYN packets where none or very few of them have completed the handshake mechanism.
8	Seagulls	A local host is receiving a number of ACK packets but has not received any other type of packet during this time window.

Continued on next page

Table 4.6 – *Continued from previous page*

No	Sound	Meaning
9	Loon	A local host is sending a number of ACK packets, while not having sent any other type of packet during this time window.
10	Cricket	A local host is receiving a number of FIN packets that are not part of a previous packet sequence.
11	Sheep	A localhost is sending a number of FIN packets that are not part of a previous packet sequence.
12	Owl	The number of incoming FIN packets received by a local host is out of proportion to the number of outgoing FIN and RST packets.
13	Horse snort	The number of outgoing FIN packets sent by a local host is out of proportion to the number of incoming FIN and RST packets.
14	Frog	A local host is sending or receiving a number of Null packets.
15	Wolf	A local host is sending or receiving a number of URG-PSH-FIN packets.
16	Beach	A local host is sending or receiving a number of packets in which the source and destination IP addresses are the same.
17	Wind on grass	A local host is sending or receiving a number of RST packets, where insufficient data has been exchanged to warrant this number.
18	Wind	A local host is sending or receiving an abnormal number of RST.
19	Snow storm	A local host is sending an abnormal number of SYN-ACK packets.
20	Walk in snow	A localhost is receiving an abnormal number of SYN-ACK packets.
21	Fire	The number of Traffic flows or IP flows is higher than the normal threshold.

4.4.8 SoNSTAR feature-to-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 4.7.

Feature construction: Example

Event 4 in Table 4.7 has the following event condition based on the three way handshake mechanism, and the function of the SYN and SYN-ACK packets in the TCP protocol: 'SYN in IPs >300 and SYN-ACK out IPs < 50 and and SYN in IPs < 1000'

This means that if a host received 300–1000 SYN packets requesting a connection while only less than 50 SYN-ACK packets were sent as a response, then the sound of thunder should be played. This sound will tell the operator that a network host is receiving a high number of requests for connection at multiple ports, but fewer than 50 open ports are responding. Sending a high number of connection requests is certainly malicious as it has to be part of heavy port scan or other malicious activities. Since less than 50 SYN-ACK packets were sent as a response, it means that some of the SYN requests are going to closed ports. Also, the RST response can be added to this event to confirm the correct response of closed ports, however, the number of SYN-ACK packets in this situation is enough to confirm the malicious behaviour.

Table 4.7 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

Continued on next page

Table 4.7 – *Continued from previous page*

No	Feature Conditions	Sound
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

4.4.9 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. 4.8 shows the SoNSTAR interactive sonification model.

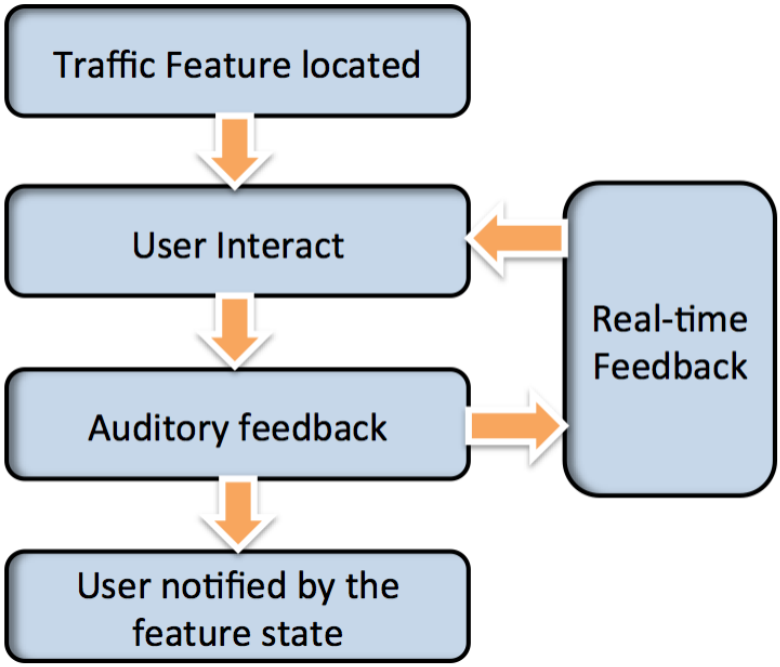

Fig. 4.8 **Interactive sonification model** showing the interactive nature of SoNSTAR.

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 of linking between physical actions and acoustic reactions produced that we have been familiar with since birth[76]. 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.

4.5 Experiment for Packet Count Concept

An experiment was conducted to explore the potential of using flag states to represent the behaviour of the cyber environment. Initially, we operated the system and visually watched the packet counts and number of flows. We noticed packet counts within the flow provided information about direction, status, and behaviour of the flows. However, the number of flows was high, making it very difficult to monitor and follow these packet counts visually (a large number of total connection flows).

We ran the system on a macOS workstation and set the time window to 5 seconds. We visited a number of websites and made the system collect all the incoming and outgoing traffic. Also, we launched some port scan attacks from a MacBook Pro system. Then we noticed the flows counts changed according to the behaviour of the traffic.

4.5.1 Results

Table 4.8 represents a sample of IP flow packet counts which are presented on the screen and saved into a log file. Packet counts presented in Table 4.8 are sorted in the following sequence: The flow number, FIN out, FIN in, SYN out, SYN in, SYN ACK out, SYN ACK in, RST out, RST in, ACK out, ACK in, PSH ACK out, PSH ACK in, URG PSH FIN in, Null in.

Table 4.9 shows a comparison between the numbers of IP flows and traffic flows within the first five time windows.

4.5.2 Discussion

Table 4.8 shows 18 IP flows. As explained previously, IP flows consist of packet counts sorted according to their flag type regardless of port numbers. These counts represent the total packet counts for each type within the IP flow. The first row in the table represents IP flow number 1. It shows the computer requested to start the connection by sending out a SYN packet to a website host and received back a SYN-ACK packet from the website, then

Table 4.8 The results of workstation traffic packet counts for the total connection sample

No.	Fo	Fi	So	Si	SAo	SAi	Ro	Ri	Ao	Ai	PAi	PAo	UPFi	Nulli
1	4	4	14	0	0	14	0	0	1006	990	110	275	0	0
2	0	0	2	0	0	3	0	0	31	21	18	27	0	0
3	3	5	7	0	0	7	0	0	193	149	47	94	0	0
4	0	0	1	0	0	2	0	0	199	190	26	77	0	0
5	0	0	3	0	0	3	0	0	77	57	26	51	0	0
6	0	0	2	0	0	2	0	0	12	6	9	12	0	0
7	1	0	1	0	0	1	0	1	444	926	4	22	0	0
8	0	0	1	0	0	2	0	0	20	8	12	18	0	0
9	0	1	3	0	0	3	0	0	208	164	53	120	0	0
10	2	2	4	0	0	4	0	4	2086	5825	22	33	0	0
11	0	0	1	0	0	2	0	0	20	8	12	18	0	0
12	1	1	0	0	0	0	0	2	1219	5944	8	12	0	0
13	0	0	0	0	0	0	0	0	3	3	4	2	0	0
14	1	1	0	0	0	0	0	0	2	2	3	1	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	640	0	1	0	0	0	1203	0
17	0	18	0	0	0	0	18	0	0	0	0	0	0	0
18	0	0	1	204	5	0	185	5	0	0	0	0	0	0

Table 4.9 IP flow and traffic flow counts

Time Window:	IP Flow Count	Traffic Flow Count
1	9	21
2	35	63
3	33	39
4	18	26
5	6	12

sent out an ACK packet to confirm this connection. Also, IP flow 1 in the table shows that 14 traffic flows were established within the IP flow during this time window. Inspection of the packet counts for each type revealed that four traffic flows were terminated within this IP flow by exchanging FIN packets in both directions. Both hosts exchanged ACK packets to confirm each flow terminated. 10 traffic flows are still active. It is clear both hosts have exchanged information as they exchanged a good number of PSH-ACK and ACK packets. Therefore, this this IP flow appears to be normal.

The second IP flow established connections using the three-way handshake mechanism twice and both are still active. Theoretically this means it is connected through two different ports (two traffic flows).

IP flow 10 shows four traffic flows established successfully. Two of them were terminated by receiving four RST packets from the website. IP flows 1 to 14 show normal flow behaviour.

IP flow 15 shows one incoming packet with Null (Zero flag). This type of packet indicates malicious activity. Usually, if only a few packets are sent they might indicate a Null port scan;if many packets are seen, it might be considered a DoS attack.

IP flow 16 shows 1203 incoming URG PSH FIN packets, and 640 outgoing RST packets; it is clear a heavy port scan is occurring and the workstation sent RST packets for each closed scanned port.

Flows 17 shows18 incoming FIN packet, and 18 outgoing RST packets in response; it is clear a port scan is going on and all the ports scanned are closed once. Theoretically, the FIN out and FIN in normal conditions should be equal, or the difference would be very small if found.

Flow 18 shows 204 incoming SYN packets, and responded by 185 outgoing RST packets; it is clear a heavy port scan is going on and the workstation sent outgoing RST packets for each closed scanned port.

Table 4.9 shows the number of IP flows was approximatley half that of the traffic flows in all the five-second time windows. The longer the time window, the lower the proportion of IP flows will be compared to traffic flows. A longer communication between any two hosts will create more and more traffic flows because of the mechanism of changing port numbers, while the number of IP flows will be always one as it depends on host IP addresses only. It can be noticed the number of flows increased and then decreased. This is because when we started connecting to a new website, a number of new flows were initiated and when the connections were broken, the number of flows dropped. Only websites with active connections will keep their flows open.

Extracting the features of these flags and representing them with sound according to the theoretical and experimental knowledge about traffic behaviour provides more understanding

to the events of the network environment. The advantage of being an interactive system is that it provides the possibility for the operator to change the conditions of events and threshold values to explore flows behaviour in the network environment. The administrator could add more features and create events according to the targeted behaviour.

4.6 Experiment for Sound Recognition and Design

Evaluating the initial design of the system sounds is important for evaluating the usability of recorded sounds to identify network events. This step is important for approving the use of recorded sounds in this monitoring system. The success of this step is important to support the use of sonification in security settings and practices to improve monitoring capabilities for situational awareness. Moreover, this evaluation can confirm that SoNSTAR is correctly generating different sounds events according to the type of attacks and traffic behaviour received. The main purpose of the experiment is to confirm that a human operator is able to recognise traffic events based on listening to recorded sounds. In addition, feedback was obtained from participants about the event-to-sound mappings design and their experience of using SoNSTAR.

4.6.1 Network Design

A macOS 10.10.5 workstation with a 3.7 GHz quad-core processor, 16 GB 1866 MHz DDR3 ECC RAM and 27-inch (2560 x 1440) display, was used as a server. SoNSTAR was used on this server to monitor its incoming and outgoing traffic.

A virtual network was created using Virtualbox and was installed on an Apple MacBook Pro running macOS 10.10.5. The computer has a 2.5 GHz intel core i7 processor, 16 GB 1600MHz DDR3 RAM and Retina,15.4-inch (2880 x 1800) display. In addition to the host machine, the Virtualbox network comprised three virtual hosts: two running Kali Linux Debian 64-bit and one running Fedora 24 64-bit in addition to the host machine.

The server and the virtual network were connected through a router. This router also provides an internet connection to both of them. Fig 4.9 shows the evaluation environment used in this experiment.

4.6.2 Participants

A call for participants was been sent through the university email system to all MSc and PhD computer science and engineering students. 19 participants (14 male, 5 female) were able to devote the time needed to participate in the study which took place in April 2016.

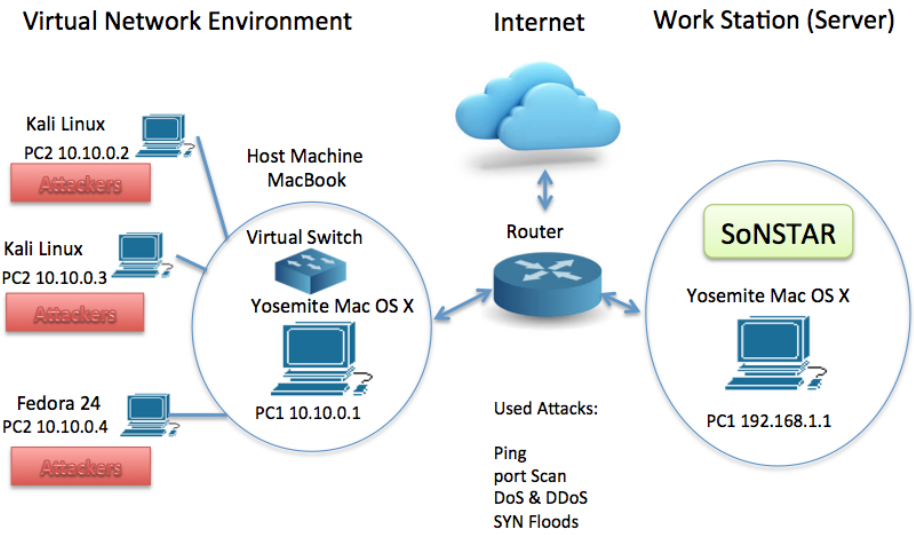

Fig. 4.9 The evaluation network setup

All 19 participants completed the study. All of the participants were aged from 25 to 45 years. The participants were PhD and MSc students at the university. 18 participants were from the Department of Computer and Information Sciences, one was from the Department of Mechanical Engineering, and the participants had good knowledge of computers and information technology and basic knowledge of computer network security.

4.6.3 Experiment design

Each participant performed a task condition which involved seven anomalous behaviours generated by penetration tests and network attacks. SoNSTAR was used to monitor the activity with the aim of detecting changes in the sounds caused by the malicious activity. The participants kept records of each sound they heard using a questionnaire table designed for this purpose (see Appendix C).

At the end of the task performance was calculated based on the number of true positive (TP), true negatives (TN), false positives (FP)and false negatives (FN), where:

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

- **FN:** represents the number of events which are incorrectly rejected. Case was positive but detected by the user as negative.

The SoNSTAR assessment used various metrics to evaluate the results of the experiment, namely TPR, or true positive rate (also commonly known as recall) true negative rate (TNR), false positive rate (FPR), false negative rate (FNR), accuracy, precision (p) and F-measure [105, 116, 124].

The TPR metric indicates the proportion of positives which are correctly detected by participants and is given by:

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

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 [105, p.1147]:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

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.

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

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

- **Traffic:** using the internet, such as playing a YouTube video.
- **Ping:** using one of any packet types for pinging.
- **Port scan:** including the four types, SYN, Null, Xmas and FIN.
- **DoS, DDoS** including first, SYN flood as type; and second, DDoS using spoofed IP addresses performed from the three machines in the virtual network.

The attacks used in this experiment in sequence from 1 to 7 are listed in Table 4.10. These attacks create different traffic behaviours which represent the seven traffic states monitored in this experiment. These attacks were performed using the Nmap scanner and Hping3 commands.

Table 4.10 Type and Sequence of Attacks Used in the Experiment

No	Attack Command	Attack Name
St.1	hping3 -F -P -U 192.168.1.23 -c 10	Xmas ping scan
St.2	nmap -sS 192.168.1.23 or nmap -sT 192.168.1.23	Nmap Scan using TCP SYN scan or TCP connect
St.3	hping3 -c 10000 -d 128 -w 64 -p 8000 –flood –rand-source 192.168.1.23	DDoS using Spoofed IP's
St.4	hping3 -c 21 -V -p 80 -s 5050 -F 192.168.1.23 or hping3 -c 3 -F 192.168.1.23	FIN scan
St.5	hping3 192.168.1.23 or hping3 -c 15 -V -p 80 -s 5040 192.168.1.23	Null scan
St.6	nmap -sX 192.168.1.23	Nmap Xmas scan
St.7	hping3 -V -S -c 1000000 -d -w - - flood 192.168.1.23	SYN DoS Flood

For example, the attack St.1 in Table 4.10 uses Hping3 to conduct an Xmas Scan. This scans type sets the sequence number to zero and sets the FIN, PSH and URG flags in the packet. The '-c' parameter sets the number of packets to be sent; in this case 10 packets are sent to host 192.168.1.23. If the target device's TCP port is closed, the target device sends a TCP RST packet in reply. If the target device's TCP port is open, the target discards the TCP Xmas scan, sending no reply. The first option of attack St.2 uses Nmap's '-sS' option

to conduct a TCP SYN scan. This scan sets the SYN flags in the packet. It is performed by quickly scanning thousands of ports per second on the host 192.168.1.23 as it never completes the TCP connections.

4.6.4 Materials

Before beginning the experiment, each participant was given an informed consent declaration to sign (see Appendix A). 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.

The training and guidelines sheet (Appendix E) included a table describing the meaning of each sound might be heard in the experiment in order to provide understanding of the expected events for the participants.

A questionnaire was given to each participant (see Appendix C). The first section elicited general participant information such as gender, level of education, speciality, department and year of study. The second section contained a list of expected sounds and check boxes in seven columns for the seven expected traffic behaviours to enable participants to tick as many sounds as they hear for each behaviour.

The third section included evaluation of monitoring workload; upon completion of each experimental task, participants completed the NASA-Task Load Index (TLX) assessment [67] 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 on a scale of 0 to 10.

The participants were then asked to evaluate the sonification in two areas *Aesthetics* and *Annoyance* from horrible to fantastic on a scale of 0 to 10. Participants could provide any extra feedback about this experiment in the final section.

4.6.5 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 the SoNSTAR task condition and where should they fill in the appropriate section. The participants' virtual network computers were switched on and some music and YouTube videos were started to generate normal traffic across the network.

The training and guidelines sheet was provided to each participant. Before starting the experimental task, participants were trained for about seven minutes in the basics of SoNSTAR and how to recognise the expected sounds. Each sound was played and named for the participant. Participants were briefed on the meaning of each sound in terms of the network event it represented. It was also explained how recognising the order in which sounds occurred was important to comprehending the traffic behaviour. Training involved demonstrating in real time one of the seven attacks used in the experiment followed by playing each sound named in the questionnaire list separately.

Each participant was provided with the questionnaire to fill in the outcomes for the task. Participants were then exposed to the SoNSTAR output for seven minutes. The participants were informed to expect one attack or penetration test each minute. At some point during each minute, the participant's workstation received one real-time attack or pen-testing activity. Participants were informed to check the boxes provided for each sound heard for each malicious activity received.

Directly after completing the task, participants were asked to answer the rest of the questions regarding the Monitoring Evaluation Task. Then they were requested to evaluate the sonification in terms of aesthetics and annoyance and to provide feedback with regards to their opinions about this experiment.

4.6.6 Results

The results obtained from the experiment were as follows.

The SoNSTAR TP, TN, FP and FN results shown in Table 4.11 are extracted from the questionnaire data. The results were calculated for the seven states to assess sound recognition as part of the situational awareness process. Based on these results, various metrics were calculated to evaluate the SoNSTAR sound design and the usability of the system. These are shown in Table 4.12.

Table 4.11 The Participants Detection Results

States	**St.1**	**St.2**	**St.3**	**St.4**	**St.5**	**St.6**	**St.7**
TP	57	88	69	74	63	86	127
TN	263	206	244	241	246	223	183
FP	3	22	3	6	1	5	7
FN	0	7	7	2	13	9	6

Table 4.12 Evaluation Metrics Results

States	St.1	St.2	St.3	St.4	St.5	St.6	St.7	Average
TPR/recall	100	92.63	90.79	97.37	82.89	90.53	95.49	92.81
TNR	98.87	90.35	98.79	97.57	99.60	97.81	96.32	97.04
FPR	1.13	9.65	1.21	2.43	0.40	2.19	3.68	2.96
FNR	0	7.37	9.21	2.63	17.11	9.47	4.51	7.19
Accuracy	99.07	91.02	96.90	97.52	95.67	95.67	95.98	95.98
Precision	95.00	80.00	95.83	92.50	98.44	94.51	94.78	93.01
F-measure	97.44	85.85	93.24	94.87	90.00	92.47	95.13	92.72

True positive rate (TPR), true negative rate (TNR), false positive rate (FPR) and false negative rate (FNR) were calculated for the seven states used in the experiment, in addition to the average rate for all states and are presented in Table 4.12. The TPR, TNR , FPR and FNR results are illustrated in Fig. 4.10.

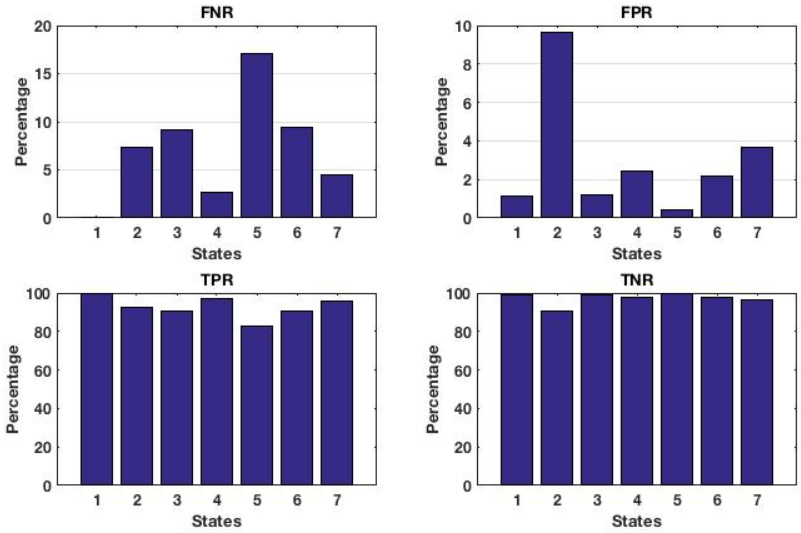

Fig. 4.10 The Detection Rates Results

Accuracy

Accuracy was calculated for the seven states used in the experiment, in addition to the average for all states and the results are presented in Table 4.12 and illustrated graphically in Figure 4.11.

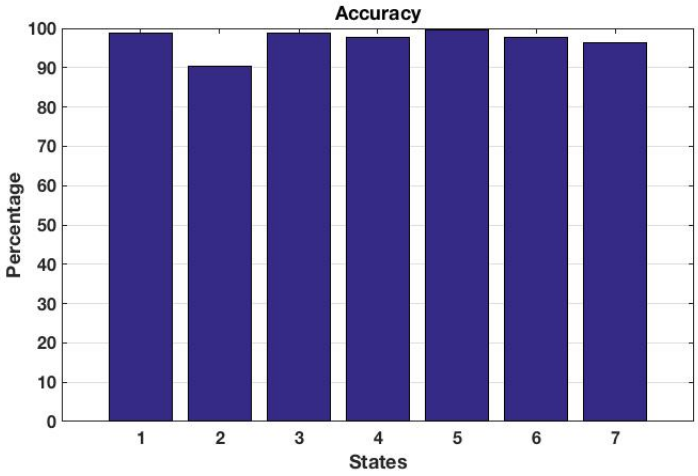

Fig. 4.11 Accuracy Results

Precision *(p)*

Precision was calculated for the seven states used in the experiment, in addition to the average for all of the states and the results are presented in Table 4.12 and illustrated graphically in Fig. 4.12.

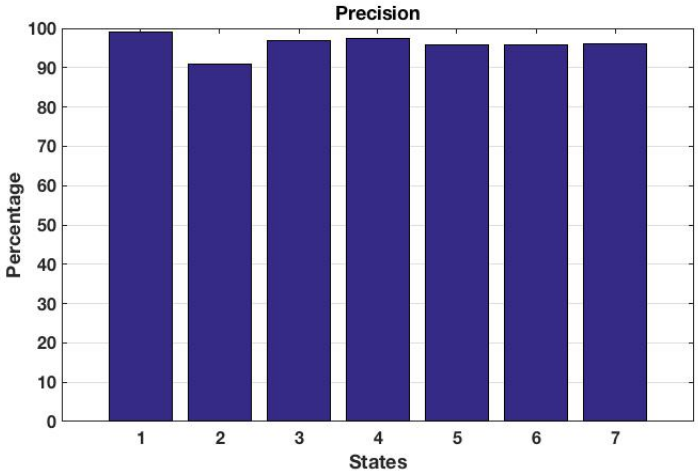

Fig. 4.12 Precision Results

TPR/Recall

The recall was calculated for the seven states, in addition to the average rate for all of the states and the results are presented in Table 4.12 and illustrated graphically in Figure 4.13.

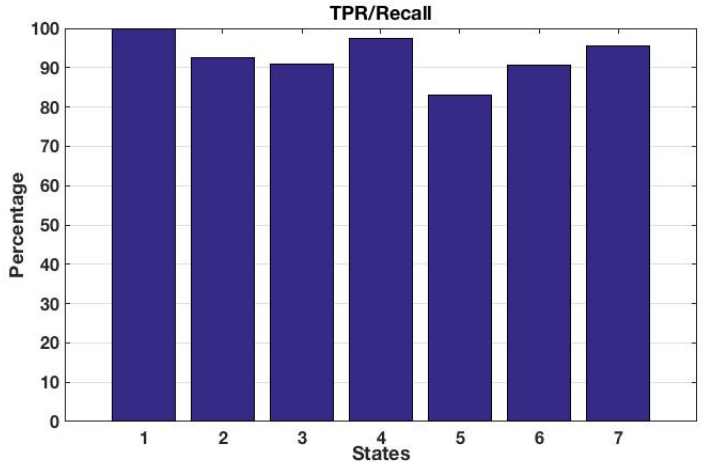

Fig. 4.13 TPR/recall Results

F-measure

The F-measure was calculated for the seven states, in addition to the average for all of the states and is presented in Table 4.12.

NASA-Task Load Index

The NASA-Task Load Index results are shown in Table 4.13.

Table 4.13 NASA-Task Load Index Results

No	Task Load Index	Rate
1	Mental Demand Rate	28.95%
2	Temporal Demand Rate	36.32%
3	Physical Demand Rate	11.58%
4	Performance Rate	85.79%
5	Effort Rate	16.32%
6	Frustration Rate	19.47%

Additional evaluation

Additional SoNSTAR evaluation results are shown in Table 4.14.

Table 4.14 Additional SoNSTAR evaluation results

No	Index	Mean Rate
1	Detection Confidence Rate	88.42%
2	Ease of Use Rate	94.21%
3	Sound Fatigue Rate	14.20%
4	Aesthetics	96.79%
5	Annoyance	13.37%

4.7 Discussion

Several attacks are used in this experiment. Each type causes different traffic behaviour. These types of attacks vary in terms of their intensity and purpose. The behaviour of each type is represented by sequential sounds in a way that shows the events of the attack. The method helps to express traffic behaviour in a sonic way rather than having to monitor the screen all the time. The purpose of this experiment was to determine first, whether the sounds used to represent the behaviour of the network could be distinguished and second, whether the sequences of sounds could be distinguished. Since situational awareness requires intelligence to be provided in real-time, it is important that the sonic representation leads to a real-time understanding of the traffic behaviour.

Table 4.12 shows that the TPR was highest for the first and fourth state behaviours where sound recognition by all participants reached 100% and 97.37% respectively. TNR was high for most state behaviours with an average of 97.04%. The FPR was acceptable although it reached 9.65% for the second state behaviour. The FNR reached a high percentage of 17.11% for the fifth state behaviour with an average of 7.19%.

Accuracy was highest in the first state where it reached 99.07% with an average of 95.98% across the seven states. Fig. 4.11 clearly shows that accuracy dropped for the second state. However, it rose again and stayed steady for the rest of the states. Precision was 95% for the

first state behaviour but reduced in the second state to 80%. Average precision was 93.01%. The F-measure was 97.44% for the first state and dropped to 85.85% for the second state behaviour. Its overall average was 92.72%. Recall started at 100% for state 1, and maintained an average of 93.81%. It was reduced in the fifth state to 82.89%.

The results show the high potential of using recorded sounds rather than MIDI tones for recognition purposes. Some systems, such as Stetho [91], faced problems due to MIDI failing to represent rich environments such as network traffic adequately. The InteNtion project [60] used MIDI messages but faced insurmountable problems in defining usable traffic-to-MIDI mappings. The use of SoNSTAR by participants inexperienced in network monitoring has shown high TPR/recall, accuracy, precision, and F-measure rates. The FPR was acceptable although it reached 9.65% for the second state behaviour mainly because the participants were mixing up the rain and rain on the roof sounds. The participants were able to recognise the sounds which they are used to such as rain, thunder, wind and fire.

The TLX scores in Table 4.13 tell us that using sound in network monitoring has demonstrated high-performance rate (85.79%) with acceptable mental, temporal, and physical demand rates. The system did not require high effort and frustration was rated low by the participants. These results support the potential of using recorded sounds in continuous monitoring tasks.

A major challenge for sonification designers continues to be that their work is often perceived as annoying, fatiguing, or both. We learn from the results presented in Table 4.14 that participants were highly confident about recognising sounds (88.42%) and considered SoNSTAR to be a user friendly system where the ease of use rate reached 94.21%. The majority of the participants were happy with the sound aesthetics (96.79%) and and very few experienced annoyance (13.37%). The sound fatigue rate was only 14.20%.These results are encouraging for the continued development of SoNSTAR for real-time monitoring of network traffic for situational awareness.

It is hard to recognise or predict normal and anomalous behaviour in computer networks. Representations of raw packet information using sound can bring administrators closer to their network behaviour and reduce the perceptual gap generated by normal IDS and other classification applications. SoNSTAR generates sounds online from real-time traffic. These sounds change according to the types of packets and their behaviour in the network.

An experiment was conducted to assess the human capability to recognise sounds generated by SoNSTAR and to evaluate SoNSTAR against the requirements of a real-time monitoring tool. The results clearly showed that the second state (a SYN scan) has the lowest TNR, accuracy, precision and F-measure which clearly indicates a drop in sound recognition although SoNSTAR has played the correct sounds for that event. The questionnaire data

show that participants varied between identifying the sounds of normal rain, rain on a roof and heavy rain, where the majority chose the normal rain sound which is considered in this experiment to represent a false positive (FP), and this increased the FPR. In addition, some participants varied between identifying the sounds of wind and wind on grass, where some participants chose both or wind sound instead of wind on grass, which generated more FPs or FNs. Most FP and FN decisions in this experiment were caused by errors in perception of the rain and wind sounds.

TPR/recall was lowest for state 5. This is because the number of false negatives (FN) increased due to some participants experiencing misperception of the wind and wind on grass sounds, where some participants chose wind or wind on grass sound instead of both. The same confusion was seen again in state 7. In addition, two participants failed to identify the fire sound as well. One of the participants was confused about the sound of crickets, which he identified in five states despite it being present in only one state.

Generally, TPR/recall and all of the other metrics show that good results were obtained from this experiment. Training took only seven minutes and many participants had heard each sound only once in the training. In addition, because ability varies between individuals some participants made only one FP mistake related to the rain sound and they were correct on all other sounds. Changing the design could be one option. However, it was considered that longer training might eliminate such confusion in future.

The sounds were generated by changes in flow behaviour on based on packet flag states and counts. Therefore, these sounds should be interpreted by the listener on the basis of their knowledge and experience of communications protocols, in addition to knowing what each SoNSTAR sound represents.

For example, when traffic with any type of packet exhibits typical traffic behaviour a type of ordinary jungle bird sound is played. Therefore, this sound will be always in the background when normal traffic is moving through the network. So, if there are bird sounds this means that there is traffic in the network. The first attack state led to the sound of wolves followed by a strong wind sound being generated. In order to translate this behaviour, we should know that the sound of wolves means many incoming URG, PSH and FIN packets while the following wind sound means that many outgoing RST packets are being sent out of normal order. It is clear URG, PSH and FIN packets are used to ping or scan closed ports which generate back as many RST packets as there are closed ports. If the URG, PSH and FIN packets were very high in number, the sounds of wind on grass would be also generated, as happened in state 6. If the number of any type of packets was large and thus generating large numbers of traffic flows or IP flows, a fire sound would be played; fire is a sign of a high-volume scan or DoS/DDoS attack.

SoNSTAR periodically gathers online flag information according to the time window set by the operator. To determine the optimal time window for traffic aggregation to allow SoNSTAR to detect malicious traffic, an initial experiment was run repeatedly with the time window being increased by a five-second increment each time. SoNSTAR was set to its default configuration and in every time window labelled each detected traffic flow as normal or malicious, storing the results in its log file. For this exercise the CAIDA DDoS Attack-2007 dataset [26] was chosen. The CAIDA dataset contains 3.5 GiB of traffic including a number of DDoS attacks. Because the only traffic recorded for the servers targeted by the DDoS attacks is the DDoS traffic, the dataset is self-labelling as any traffic relating to other hosts is known to be normal. It was found that the rates of DDoS detection were high for all time windows (see Table 4.15). The results show that a 35 s time window provided the best accuracy and precision. However, both the 15 s and 20 s time windows showed an accuracy of 99.8% and precision of 99.96%, which make them suitable to be used as well.

Table 4.15 SoNSTAR Evaluation Metrics for CAIDA DDoS Dataset

Time window	**FPR**	**Recall**	**ACC**	**Precision**
5 s	0.03%	97.8%	97.8%	97.92%
10 s	0.02%	98.4%	98.4%	98.92%
15 s	0.01%	99.6%	99.6%	99.94%
20 s	0.10%	99.8%	99.8%	99.96%
25 s	0.01%	99.7%	99.8%	99.96%
30 s	0.01%	99.4%	99.4%	99.9%
35 s	0%	100%	100%	100%

In Chapter 5 an experiment is presented that compares SoNSTAR sonification to an IDS system (Snort) to test human ability to comprehend sounds and how practical SoNSTAR is in raising or maintaining situational awareness levels. SoNSTAR's current design is able to extract TCP, UDP and ICMP protocol packet information. Since the ICMP ping packets usually have a relation to TCP protocol events, extra development was conducted to the system to include ICMP ping events. 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 events.

4.8 Summary

A novel interactive sonification system for network monitoring (SoNSTAR) based on traffic flow and IP flow features has been developed to support real-time monitoring to increase network administrators' situational awareness. This chapter describes the design of SoNSTAR and the features list used to trigger the sounds generated. The sound mapping mechanisms combined with the introduction of the concept of IP flow allowed the huge volumes of network traffic to be reduced to manageable sizes for sonification.

Two initial evaluations (Sections 4.5 and 4.6) are described and they are used to support the primary design of SoNSTAR. The evaluations show that the packet feature counts change according to the traffic behaviour and the sounds generated are easy to learn and recognise and SoNSTAR succeeded in transforming the network environment into a soundscape environment.

Chapter 5

Sonification of Network Flow Events for Monitoring and Situational Awareness

5.1 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 intrusion detection software), and 3) audio and visual feedback together (SoNSTAR and Snort). Snort was chosen to be used in this experiment because, it is a de facto standard (being used inside many current popular network security tools) which makes the evaluation more general [78, 141].

5.1.1 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. The 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 an Apple 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 (see Figure 5.1).

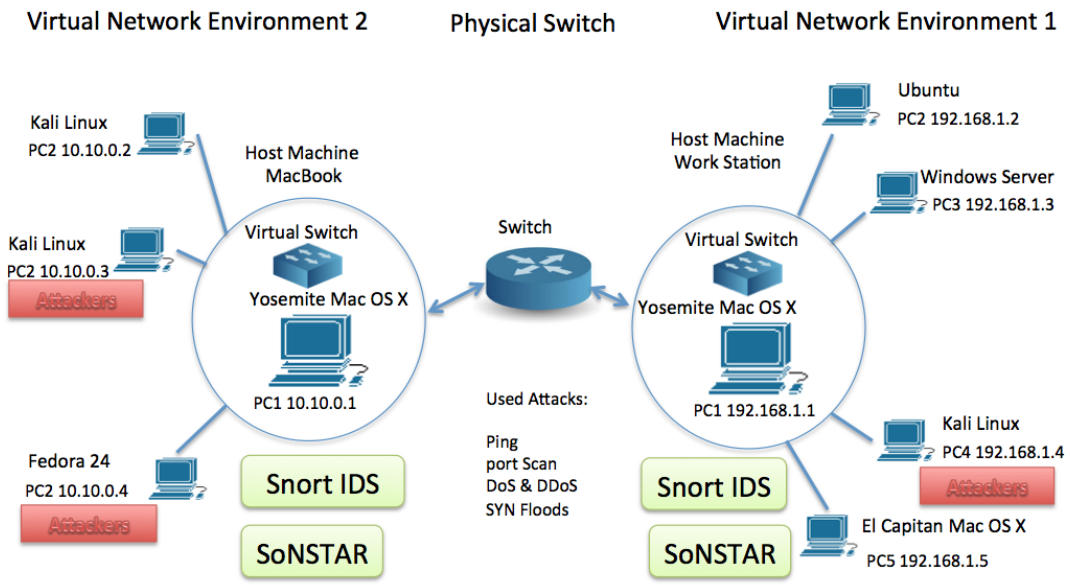

Fig. 5.1 Illustration of the Virtual Network Environment.

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.

5.1.2 Participants

A call for participants was been sent through the university email system to all MSc and PhD computer science and engineering students. 16 students responded to the email and 10 participants (7 male, 3 female) were able to devote the time needed to participate in the study which took place in September 2016. All 10 participants completed the study. All of the participants were aged from 25 to 45 years and were PhD and MSc students at the university (8 from the Department of Computer and Information Sciences). All participants had good knowledge of the use of computers and information technology and general knowledge about computer network security.

5.1.3 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 three or four attacks out of seven overall.

The participants were assigned to use either Snort or SoNSTAR first (five participants use each system first) 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). Other metrics were then calculated using the same method as described in Chapter 4.

Snort's detection rules were set to the defaults provided by the `snort.conf` file [150]. The Snort default ruleset provides a basic set of network intrusion detection rules developed by the Snort community which allow the detection of typical probes and attacks such as stealth port scans, DoS/DDoS attacks, CGI attacks and buffer overflows.

SoNSTAR was set to the sound mapping presented in Table 4.7. 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, the Nmap scanner and Hping3 commands. The folder '`examples`' in the project repository [41] contains examples audio files of SoNSTAR sonifications of the activities used in this experiment as follows:

1. S1 Audio. **Normal traffic behaviour.** SoNSTAR normal events sounds audio file.
2. S2 Audio. **FIN behaviour.** SoNSTAR FIN scan audio file. The scan performed using hping3.
3. S3 Audio. **Xmas behaviour.** SoNSTAR heavy Xmas scan audio file. The scan performed using Nmap.
4. S4 Audio. **NULL behaviour.** SoNSTAR low NULL scan audio file. The scan performed using hping3.

5. S5 Audio. **NULL behaviour.** SoNSTAR heavy NULL scan audio file. The scan performed using hping3.

6. S6 Audio. **SYN behaviour.** SoNSTAR heavy full connection SYN scan audio file. The scan performed using Nmap.

7. S7 Audio. **Ping behaviour.** SoNSTAR SYN-Flood-DOS audio file. sounds of SYN flood attack behaviour for denial of service purpose; performed using hping3.

5.1.4 Materials

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

The training and guidelines sheet (see Appendix F) 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.

A questionnaire was given to each participant (Appendix D). The first section elicited general participant information such as sex, 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 [67] 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.

5.1.5 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.

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.

5.1.6 Results

Several results are extracted from the questionnaire data as follows.

The results for the three conditions are shown in Table 5.1 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.

Table 5.1 The Detection Results

Metrics	**Snort**	**SoNSTAR**	**Snort & SoNSTAR**
TP	31	33	30
TN	31	33	38
FP	7	4	2
FN	0	0	0

The metrics calculated from the base variables are shown in Table 5.2.

Table 5.2 Evaluation Metrics Results

Metrics	**Snort**	**SoNSTAR**	**Snort & SoNSTAR**
TPR/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%

TPR/Recall was calculated for the three conditions used in the experiment and the results are presented in Table ch5:table2 and illustrated graphically in Fig. ch5:fig2.

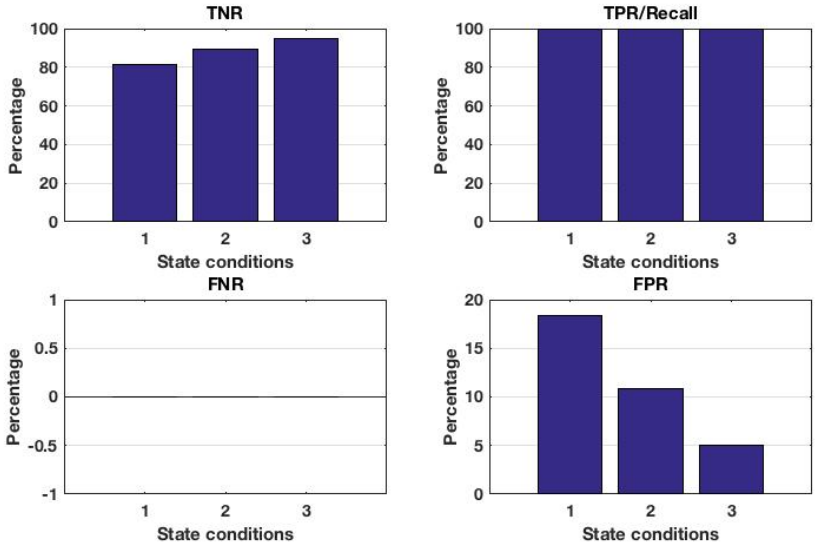

Fig. 5.2 The Detection Rates Results. State conditions were Snort, SoNSTAR, Snort + SoNSTAR respectively.

Accuracy was calculated for the three state conditions used in the experiment and the results are presented in Table ch5:table2 and illustrated graphically in Fig. ch5:fig3.

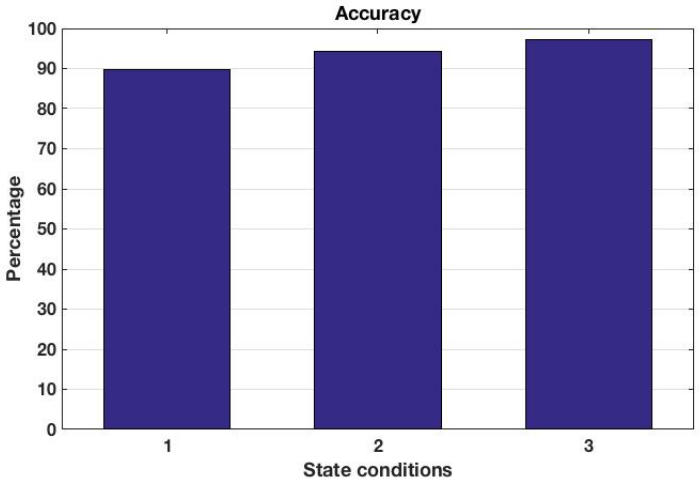

Fig. 5.3 Illustration of Accuracy Results

Precision was calculated for the three state conditions used in the experiment and the results are presented in Table ch5:table2 and illustrated graphically in Fig. ch5:fig4.

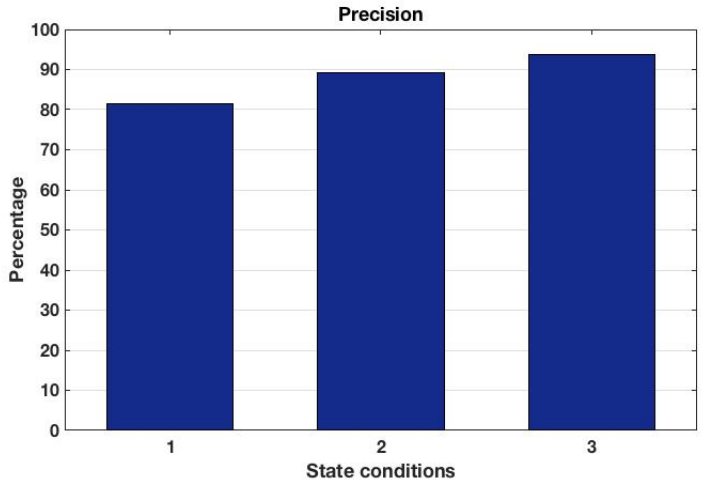

Fig. 5.4 Illustration of Precision *(p)* Results

F-measure was calculated for the three state conditions used in the experiment and the results are presented in Table ch5:table2 and illustrated graphically in Fig. ch5:fig5.

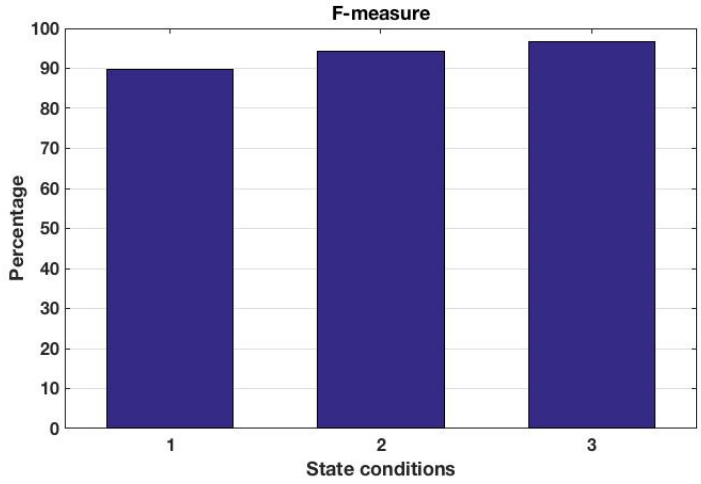

Fig. 5.5 Illustration of F-measure Results

NASA-Task Load Index results

The mean NASA TLX results for the ten participants are shown in Table 5.3.

Table 5.3 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%

Additional evaluation results

Additional SoNSTAR evaluation results are shown in Table 5.4.

Table 5.4 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%

Table 5.5 shows participants' opinions about whether using Snort and SoNSTAR alone or together would be best for monitoring.

Table 5.5 Additional SoNSTAR evaluation (preference results)

Index	Snort	SoNSTAR	SoNSTAR & Snort together
Best to use	10%	30%	60%

Table 5.6 shows participants' opinions about Snort and SoNSTAR from horrible (H) to fantastic (F).

Table 5.6 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%

5.2 Discussion

One of the purposes of this experiment is to compare the use of a sonification monitoring system with a visualisation monitoring system to determine what value for the operator can be added. A second purpose was to find out whether using visualisation and sonification systems together increases the efficiency of the monitoring process. One of the questions asked is whether the use of sound will facilitate the monitoring process or not.

Table 5.2 shows a maximum recall of 100% for the three state conditions (Snort, SoNSTAR, Snort + SoNSTAR respectively). 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. Figure 5.3 clearly shows that accuracy improved when using sonification.

Precision of recognition was highest when using both Snort and SoNSTAR together at 93.75%. SoNSTAR maintained higher precision again compared to Snort at 89.19% and 81.58% respectively. Figure 5.4 clearly shows that precision also improved when using sonification.

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. Figure 5.5 shows that the F-measure rate improved as well when using sonification.

The results show that using sonification (SoNSTAR) and visualisation (Snort) together achieved better results than using each one alone. This indicates that integrating sonification and visualisation techniques increases the monitoring efficiency. In general, SoNSTAR has

achieved better results than Snort as the changes in sound notified the participants of the change in behaviour. This means sonification has improved monitoring capabilities and shows evidence of the potential of sonification in improving network security monitoring capabilities.

These experimental results clearly show improvements in monitoring when using sonification compared to the visual method only. Although the TPR/recall was 100% for the three state conditions, we can still see improvements in the accuracy, precision and F-measure scores for the sonification condition. 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 in a very good way in a short time.

Comparing The TLX scores in Table 5.3 shows the advantage of using sonification in monitoring. Mental demand rate shows the mental and perceptual workloads required when using sonification were less than when using Snort which relied on monitoring messages and alerts that appeared on the screen (45% vs. 58%). The temporal demand rate indicates the pressure participants felt due to the pace of the monitoring task. The scores reported by participants using Snort were approximately double those when using SoNSTAR (31% vs. 65%). This means that the participants were more comfortable when using sonification for monitoring. Both systems required low physical activity. Performance rate indicates how successful the participants thought they were in detecting the traffic behaviours. The participants were more satisfied with their results in the sonification condition (92% vs. 82%). Effort rate indicates how hard participants had to work to complete the monitoring task. The effort rate for Snort was approximately double that of SoNSTAR (19% vs. 41%). This indicates using sonification can ease the monitoring process. Frustration rate indicates how stressed, irritated, or annoyed participants felt as opposed to feeling content and relaxed. SoNSTAR elicited a lower frustration rate than Snort as it does not require concentrating and waiting while staring at a screen (36% vs. 71%). Having to watch a screen can lead to feelings of stress at the prospect of missing any reported changes in behaviour. The sound has the advantage of notifying the participants of changes in behaviour as soon as they happen. This advantage of using sonification can be better addressed when monitoring for long periods of time. We believe that the frustration rate will be more pronounced if the two systems are used for a long period of time at least an hour. In that case, the advantages provided by the sound will become more user-friendly as the user does not need to look at the screen all the time.

Table 5.4 illustrates that both systems showed high confidence of the participants in detecting traffic activities as well as being easy to use. However, the visual fatigue rate was higher than the sound fatigue rate which also indicates more advantage can be obtained

when using sonification in monitoring. Table 5.5 results tell us that targeting sonification and visualisation techniques in monitoring would produce better monitoring systems for the users. According to Table 5.6, most of the participants consider Snort and SoNSTAR are good systems to use.

The most remarkable feedback was that a participant asked whether it was possible to add a visual panel showing the name, colour and image of what is generating the sound as 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.

SoNSTAR is designed to be used by professional network users who have good knowledge about protocols, packets, and traffic flows behaviour. Otherwise, users have to be trained for a longer time to understand the value of each sound for comprehension because each behaviour could be based on multiple sounds. For example, a single ICMP ping packet would produce a woodpecker sound. This sound would tell the user that an ICMP ping has happened. If repeated, this might be considered a further scan.

Some other 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 packets 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 nature 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.

Another feature that SoNSTAR possesses is that it generates log files which could help any user to learn and confirm the reasons for each sound and to evaluate any theoretical event and ideas of a new feature. This 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 significantly compared to using Snort, a visual tool and this would be more obvious if the monitoring was for long hours.

5.3 Summary

The current design purpose is monitoring total network traffic. As a further step to evaluate SoNSTAR against IDS systems and investigating the advantages that could sonification bring to raise security situational awareness level, this experiment was performed.

This study indicated that using sonification improved the monitoring process, even for people who have only basic knowledge about network monitoring. Using sound reduced the overall mental workload. Participants were able to recognise and comprehend behaviours and decide which attack was performed which proved the human mind could learn quickly about the network environment in a way that would result in increasing the security situational awareness. Although the system could be evaluated manually by comparing against the log files, this experiment evaluated the practicality of using sonification in live monitoring tasks. The results suggest that using SoNSTAR to explore new event and features would bring benefits to IDS systems and network monitoring in general and for situational awareness.

A contribution of this chapter was to reduce the complexity of huge volumes of traffic in order to be comprehensively sonified by using IP flow in detecting network behaviour, especially vertical flow behaviours. Also the sound mapping of the network events based on packet type counts has not been seen before. The next chapter shows how SoNSTAR was applied to the discovery of horizontal flow behaviours. Vertical flow behaviour occurs when a single host receives many flows across 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.

Chapter 6

Sonification Approach to Support IDSs to Detect and Learn about Botnet Behaviour

With the increase of computer network attacks through botnets, the majority of networks of all sizes are at risk [31]. Security leaders are looking for new ways to improve their current security monitoring tools for efficient botnet traffic detection. Signature and anomaly-based IDS technologies use advanced techniques (such as neural networks) to detect and block attacks effectively. However these IDS and visualisation systems do not include the protocol flow granularity required to understand network events inside an environment; they just report what happened but not why.

Raising the security situational awareness of the user is high demand in order to quickly react to situations which require real-time solutions, intelligence, and human intuition to deal with botnets and other malicious activities. These technologies with their different levels of data granularity, sonification techniques and integrations with visualisation have not been subjected to much study with users.

This chapter explores how SoNSTAR may be used alongside traffic log files to enable the user to target and detect botnet behaviour and reveal important aspects of botnet behaviour.

6.1 Introduction: Botnets

Network administrators commonly use a combination of intrusion detection system (IDS) software and sensors that inspect traffic on the network, and wait for anomalous events to occur. Intrusions are defined as ‘attempts to compromise confidentiality, integrity, or

availability of data, or to bypass the security mechanisms of an IT system' [2, p.147]. Network security monitoring has become a crucial task in protecting organisational infrastructure from today's threats which have seen intrusions and attack patterns becoming hidden and more complex [31].

In 2015, IBM analysts reported a number of types of attempt to break into networks and organisational infrastructures, such as exploiting 'a vulnerability to inject command code into software, exploiting a backdoor, or bombarding a system with random passwords in hopes that one will work' [21, p.3]. Their report declared that the majority of networks of all sizes are at risk and that 'CISOs and security leaders are now looking for fundamental ways to influence and improve both their own programs and [previously] established best practices' [21, p.14].

This chapter looks at the problem of detecting botnet traffic which rely on vertical and horizontal traffic behaviours. A botnet is a network of remotely controlled devices, or 'bots', such as personal computers and smartphones whose security has been breached and control access given to a third party. The botnet controller directs the activities of the bots through messages sent via standard network protocols. Botnets are used for various malicious purposes such as conducting distributed denial of service (DDoS) attacks, spreading spam, spying, and stealing personal information [172]. They propagate over legitimate communication connections and, because an individual bot may only send a few packets to the host under attack, the volume of traffic looks normal.

Behind every attack is an underlying motive, and knowing what it is might allow administrators to anticipate attacks that might be deployed against their networks [148]. Axelsson and Sands suggested that in 'dealing with the more imaginative threats, a human operator needs to be in the loop and in order to be effective there should be tool support that enables her to quickly gain an understanding of the situation' [5, p.26].

6.1.1 Related Work

Existing IDS technologies rely on a variety of techniques to detect botnets, including identifying repetitions of requests, statistical methods [146, 171] and entropy detection [86], and all such techniques tend to 'collect flow information from bots to depict their behaviour' [87, p. 976].

Data mining for botnet detection aims to identify useful patterns to discover regularities and irregularities in massive data sets. Since individual flows in a botnet attack are not malicious by themselves unless they are found to be part of a series of synchronised flow connections, a wide range of data mining techniques including classification, correlation, clustering, aggregation, and statistical analysis is used for knowledge discovery about network

flows [139]. Machine learning has been widely used in the detection of botnet methods. Machine learning is a category of artificial intelligence that aims to advance systems with the intelligence to learn from past experience [110]. For example, Ranjan et al. [127] introduced a machine learning method to detect botnets using connectivity graph-based traffic features derived from historical network data.

Machine learning techniques come in three different kinds: First, supervised learning which works with labelled data; second, unsupervised learning used unlabelled data; third, reinforcement learning systems use both supervised and unsupervised techniques to take control of their own learning. Reinforcement learning (RL) is inspired by behavioural psychology with regard to how software agents need to take action in an environment in order to increase rewards [106, 110]. Dejmal et al. [43] used RL for proactive assessment in peer-to-peer networks to assess the vulnerability to unknown future network attacks. The technique has significant impact potential.

A challenge that faces all detection techniques is validating them on real networks which vary from the test environments in which they were developed [87].

To-date, using visualisation techniques to support botnet detection has received only modest attention. Seo et al. [137] proposed a security visualisation tool called CCSvis to target botnet behaviour based on Domain Name System (DNS) traffic. The system presents visualisations of traffic using cylindrical coordinates to enable a human operator to identify botnet behaviours and patterns. Thus, detection is a cooperative activity involving both human and machine.

Kim et al. [90] also visualised DNS traffic with the aim of detecting botnets before they start carrying out their attacks. They defined four patterns of graphs as botnet signatures which can be identified by the human operator. Experimental results suggested that visualisation could be used to detect both known and unknown botnet types.

Visual Threat Monitor [139] is a flow-based system which combines data mining and visualisation to enhance botnet traffic detection. Its visualisation method uses processed and selected data rather than raw data and the outputs consist of correlations, statistical analysis, clustering, aggregation, and visualisation.

While some network sonification work has been reported [10, 53, 59, 91, 159, 165, 167] the technique has not yet been applied specifically to botnet detection. Below we show how the SoNSTAR system may be used to complement an existing IDS by sonifying network traffic in such a way as to enable botnet behaviour to be detected and identified by a human operator without the use of any botnet detection algorithms.

6.2 Botnet Sonification Using TCP

Sonification has the potential to assist in the discovery of patterns of botnet network activity and relationships between seemingly disparate security events, though little has been done to leverage sonification technologies in current practice. SoNSTAR was designed to fit the work practices and operational environments of network security monitoring analysts in order to raise their situational awareness through reflecting human understanding in the monitoring process. The solution starts by extending SoNSTAR to enable the operator to explore a network's botnet traffic patterns. Publicly-available labelled botnet datasets were used to demonstrate the technique.

6.2.1 Characteristics of a Botnet

TCP botnet traffic has certain characteristics that can be distinguished amongst legitimate traffic as follows [15, 46]:

1. Botnet lifecycles go through the same five stages. The first stage is infection and propagation where the botmaster infects new targets such as computers or servers to became bots. Propagation mechanisms refer to the method used to expand and search for new machines. They consist of horizontal scans, vertical scans, coordinated scans and other sophisticated propagation methods. A vertical scan is described as scanning a single host across a defined range of ports. Horizontal scans are where a single port is scanned across a defined range of hosts.

   The second is rallying, where a bot connects to the C&C server or the bot receives updates such as a list of C&C IP servers. The third is command and report, where the bot connects to the C&C server to receive commands and to send its activity reports. The fourth is 'abandon' where a bot becomes unusable. The fifth is securing, where the botmaster tries to conceal its bots from security detection systems.

2. The command and control (C&C) mechanism has three architectures. The first is centralised, where the botmaster communicates with bots through a central C&C server. The second form is decentralised, where bots also act as C&C servers based on a peer-to-peer (P2P) network model. The third type is hybrid, where the botmaster can use any applicable protocols and architectures to implement its model.

3. Botnets perform several malicious activities depending on their size (large- or small-scale) such as DDoS and spamming.

However, botnets are a rapidly evolving phenomenon that is still not well understood. Moreover, bots mostly connect with C&C servers or other infected nodes which act as C&C servers using normal traffic communication patterns which are repeated again and again. Bots are relatively consistent in the repetitive communication mechanisms employed between them when they are using a P2P network model. Bots may scan the network to collect information that may help the botmaster to prepare for future attacks such as infecting other devices. Also, bots might be part of a botnet network launching an attack such as a DDoS where they continuously communicate with an external host using similarly repetitive communication traffic patterns. Accordingly, SoNSTAR was extended to extract features and to enable users to create events and mappings to explore and identify bot behaviours.

6.2.2 Exploring Traffic for Botnet Detection

Exploring traffic aims to recognise useful events so as to recognise botnet behaviour. Since botnet traffic does not have a specific behaviour, we aimed to create more features that allow the discovery of events which might be part of a botnet behaviour. This method allows the user to assign sounds to different events based on his/her knowledge and experience and the literature on botnets.

Botnets exhibit stealthy behaviour by several methods including distributed behaviour, parallel behaviour, repetitive behaviour and stealth scanning. Stealth scanning is described as using vertical or horizontal scans with low frequency to avoid detection [169]. Parallel flow behaviour occurs when a single host establishes flow connections with several network hosts on several ports. Distributed flow behaviour is where a local host receives several flows from different external hosts on several ports. Repetitive flow behaviour is where repetitive flow patterns are observed and which are caused by bots repetitively carrying out the same task (performed automatically or on a schedule) over the internet [57]. Although the previous SoNSTAR design is capable of addressing various traffic behaviours, horizontal, distributed, parallel and repetitive stealth behaviours were not addressed.

Since, on the face of it, botnet flows resemble normal traffic, additional features need to be considered to support the recognition of botnet activity. The method consists of using sonification to monitor events which are suspected as evidence of botnet activity. The operator then reviews the log files corresponding to these suspicious events and creates a pattern based on IP flow features to match the selected events. This pattern can then be associated with a specific sound in SoNSTAR allowing any occurrences to be monitored. For example, in normal traffic behaviour, it is virtually impossible to find identical IP flow patterns repeated within a single time window.

Botnet activity could consist of several different combinations of traffic events. It is left to the human operator to explore and decide which events might be part of botnet behaviour. The correlation of events is useful for monitoring botnets by looking botnet characteristics. This approach is very effective when performing real-time monitoring. The human mind correlates events to understand situations based on event sounds, sequence, occurrence, and other factors such as the nature of the network and the motivation of expected attacks. There is no specific rule to be used to recognise botnets but a human can look for various events and behaviours and tune thresholds according to his accumulated knowledge and understanding of botnet activity such as stealth and repetitive behaviours, which indicate botnet activity.

6.2.3 Extended SoNSTAR design

To address the problem of botnet detection SoNSTAR's design was extended as follows.

Feature Extractor — Selected TCP Parameters

SoNSTAR works by selecting combinations of IP traffic features and mapping these feature combinations to discrete sounds in its soundscape. Table 6.1 shows 29 of these mappings (see the project repository [41] for further details). Four new arrays were implemented to collect more features that facilitate the targeting of repetitive parallel, horizontal and distributed flow behaviour.

Table 6.1 Feature-to-Sound Mappings

No	Feature Conditions	Sound
1	SYN in IPs <30 and SYN ACK out IPs >0 and ACK in IPs >0 and RST out IPs <10	Forest bird
2	SYN in IPs >10 and SYN in IPs <30 and PSH ACK out IPs == 0 and RST out IPs >0	Rain on roof
3	SYN in IPs >8 and SYN ACK out IPs <4 and PSH ACK out IPs <50	Rain on roof
4	SYN in IPs >300 and SYN ACK out IPs <20 and PSH ACK out IPs <300	Thunder
5	SYN out IPs >10 and SYN ACK in IPs <3 and PSH ACK out IPs == 0 and RST in IPs >0	Rain
6	SYN out IPs <30 and SYN ACK in IPs >0 and PSH ACK out IPs >0	Forest bird

Continued on next page

Table 6.1 – *Continued from previous page*

No	Feature Conditions	Sound
7	ACK in IPs >1 and RST out IPs >0 and the rest of IP flow feature == 0	Seagulls
8	ACK out IPs >1 and RST in IPs >0 and the rest of IP flow feature == 0	Loon
9	FIN in IPs >9 and FIN in IPs >SYN out IPs and FIN in IPs >SYN in IPs and F 4 >10 and PSH ACK out IPs <2 and PSH ACK in IPs <2	Cricket
10	FIN in IPs <50 and (FIN in IPs <= SYN out IPs or FIN in <= SYN in IPs)	Forest bird
11	FIN out IPs >9 and FIN out IPs >SYN out IPs and FIN out IPs >SYN in IPs and FC 3 >10 and PSH ACK out IPs <2 and PSH ACK in IPs <2	Sheep
12	FC 7 >9 and PSH ACK out IPs <5 and PSH ACK in IPs <5	Owl
13	NULL in IPs or NULL out IPs >0	Frog
14	URG PSH FIN in IPs or URG PSH FIN out IPs >0	Wolf
15	LAND in IPs or LAND out IPs >0	Beach
16	RST in IPs >30 and ACK in IPs <100 and PSH ACK out IPs or PSH ACK in IPs <2	Wind on grass
17	RST out IPs >30 and ACK out IPs <100 and PSH ACK out IPs or PSH ACK in IPs <2	Wind on grass
18	FC 1 >4 and PSH ACK out IPs or PSH ACK in IPs == 0	Fountain
19	FC 2 >4 and PSH ACK out IPs or PSH ACK in IPs == 0	Heavy rain
20	RST out IPs >30 and FC 5 <FC 14 and ACK out IPs <5 and PSH ACK out IPs <2	Wind
21	RST in IPs >5 and FC 6 <FC 15 and ACK in IPs <5 and PSH ACK in IPs <2	Wind
22	Flow Counter >1000	Fire
23	IPs Flow Counter >600	Fire
24	Src addr1 A count >200	Mosquito
25	Src addr1 A count >50 and Identical packet counts1 >250	Mouse squeaking
26	Dst addr2 A count >200	Bee Colony

Continued on next page

Table 6.1 – *Continued from previous page*

No	Feature Conditions	Sound
27	Dst addr2 A count >50 and Identical packet counts2 >250	Rats (multiple) squeak
28	Src addr count3 >85 and Dst port count3 >95	Spring Peeper
29	Dst addr count4 >85 and Dst port count4 >95	Grass hopper

The first array collects features which are used to identify any local hosts that attempt to perform parallel repetitive flows within the local network during a time window. Fig. 6.1 illustrates the features collected to target local source IP addresses which perform internal network parallel repetitive behaviours. The first column in the array contains a local source host, the next column contains a destination port number, and the next column contains the number of packets sent from the local source host to the destination port number. The features collected by the feature extractor into array 1 are listed in Table 6.2.

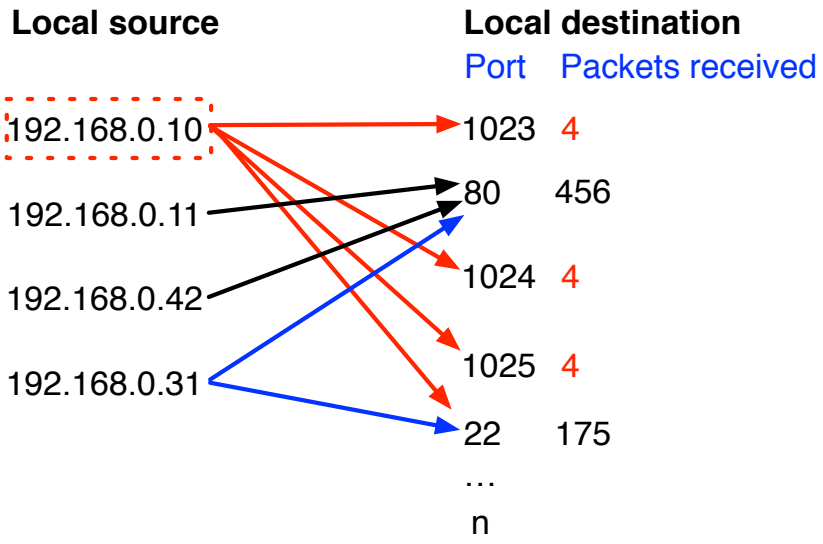

Fig. 6.1 Features for targeting the behaviour of local parallel repetitive flows. Local host 192.168.0.10 is repeatedly sending the same four packets to certain ports on multiple destination hosts on the network.The features collected are: list of local source hosts, destination port numbers, count of packets sent to each port.

The second array collects features that help to identify any local hosts that experience distributed repetitive flow behaviour from external hosts during a time window. Fig. 6.2 illustrates the features collected to target local IP addresses which have experienced distributed

Table 6.2 The features collected in array 1

Element	Label	Description
1	Src addr1 A	Local host A (Source IP) has sent packets to another local host B during current time window period.
2	Dst port1 B	The destination port number on local host B has received one or more packets local host A identified in element 1.
3	Packet count1	The number of packets sent from local host A to the local host B through the destination the port number identified in element 2.

repetitive flow behaviour. The first column in the array contains a local destination host, the next column contains a destination port number, and the next column contains the number of packets received from the external source host by the local destination hosts through the destination port number. The features collected by the feature extractor into array 2 are listed in Table 6.3.

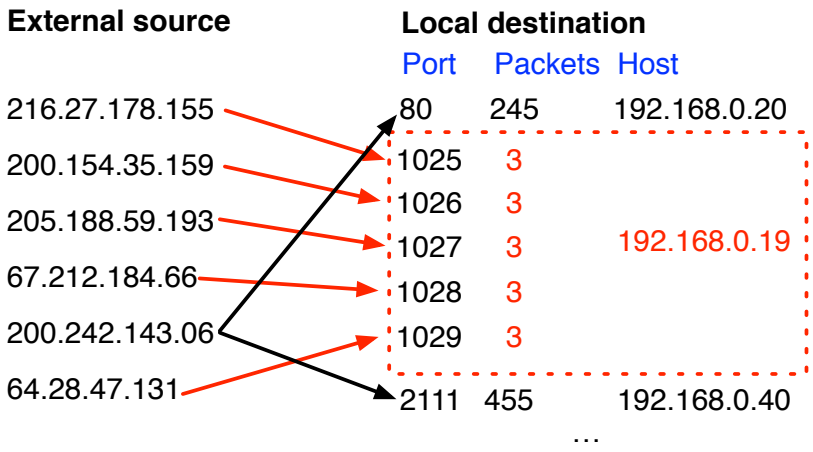

Fig. 6.2 SoNSTAR Features for targeting the behaviour of incoming distributed repetitive flows. Multiple external hosts are targeting ports on local hosts with the same three packets. The features collected are: list of local destination addresses, destination port numbers, count of packets sent to each port.

The third array collects features that help to identify any local hosts receiving distributed flows (such as a hidden scan behaviour and malicious distributed flows as legitimate queries) from external hosts during a time window. Fig. 6.3 illustrates the features collected to identify local hosts receiving flows from external hosts such as stealth distributed scans or other repetitive distributed horizontal activities. The features are collected for all local

Table 6.3 The features collected in array 2

Element	Label	Description
1	Dst addr2 B	Local host B has received one or more packets or from external hosts during the current time window period.
2	Dst port2 B	The destination port number on local host B identified in element 1 which has received one or more packets from an external host.
3	Packet count2	The count of packets received through the destination port identified in element 2.

destination host in every time window. The features collected by the feature extractor into array 3 are listed in Table 6.4.

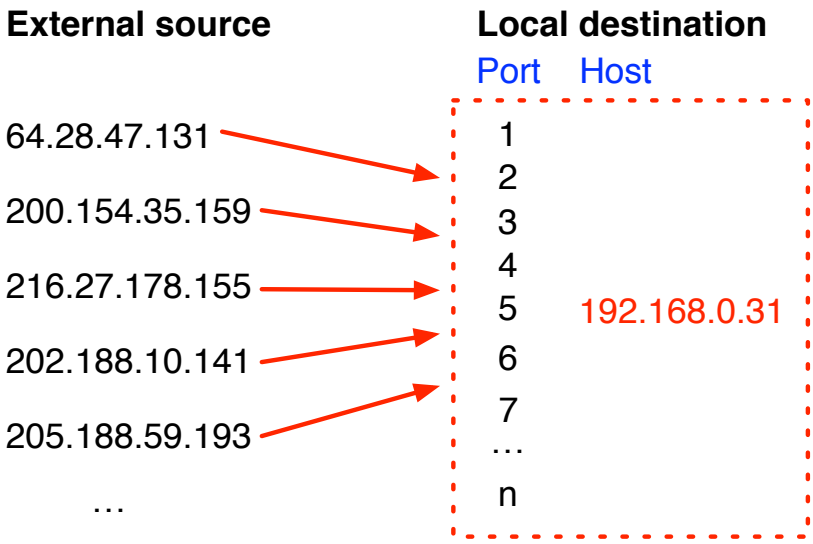

Fig. 6.3 Features for targeting incoming horizontal flow scan. Multiple external hosts are targeting ports on a single local host. The features collected are: list and count of local ports, list and count of source addresses.

The fourth array collects features that help to identify any local hosts attempting to perform local horizontal and parallel activity within the local network during a time window period. Fig. 6.4 illustrates the features collected to target internal network horizontal scans, and the features are listed in the Table 6.5.

Features combiner

At the end of each time window, the new features extracted using algorithms 1 to 4, together with the IP flow features and traffic flow features are passed to SoNSTAR's feature combiner.

Table 6.4 The features collected in array 3

Element	Label	Description
1	Dst addr3	Local host B has received one or more packets from a local and external hosts.
2	Src addr3 list	List of source addresses that have sent packets to local host B (element 1).
3	Src addr count3	The count of source addresses in the list from element 2.
4	Dst port list3	The list of destination ports on local host B which received one or more packets from sources in the hosts list.
5	Dst port count3	The count of destination ports in the list identified in element 4.

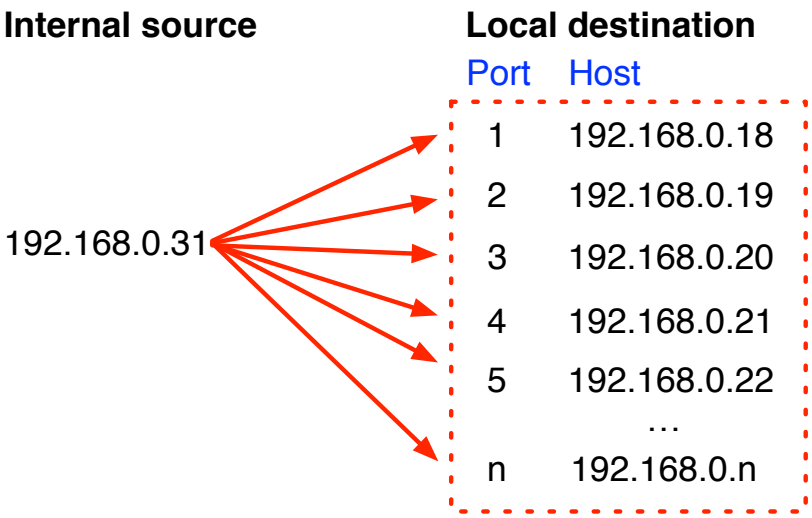

Fig. 6.4 Features for targeting local horizontal scan and parallel flow activities. The features collected are: list and count of destination hosts, list and count of destination ports.

The feature combiner uses the IP flow and traffic flow features to obtain newly discovered combinations to be used to create new events.

Since the existing SoNSTAR design already has eight combinations, the design has been extended by adding seven new features as shown in Table 6.6. These feature combinations are created as a short hand to make it easier for the user. For example, SYN out IPs and FIN in IPs are related to each other in the TCP protocol as SYN out starts the connection and FIN in ends the connection. During use and experimentation with SoNSTAR, many times it was necessary to add these feature counts to each other to create new events. So a separate feature

Table 6.5 The features collected in array 4

Element	Label	Description
1	Src addr4	Local source host A has sent one or more packets to a local destination host.
2	Dst addr list4	The list of local destination hosts that have received one or more packets from local host A.
3	Dst addr count4	Count of destination hosts in the list from element 2.
4	Dst port list4	The list of destination ports of the local hosts from element 2 which received one or more packets from local host A.
5	Dst port count4	Count of destination ports in the list from element 4.

(FC9, Table 6.6) was constructed which can be used directly without the need to explicitly combine both features every time.

Table 6.6 Feature Combinations

Feature Combination	Description of Feature
FC 9	Result of (SYN out IPs) + (FIN in IPs)
FC 10	Result of (ACK in IPs) − (ACK out IPs)
FC 11	Result of (FIN in IPs) + (FIN out IPs)
FC 12	Result of (PSH ACK in IPs) − 1
FC 13	Result of (ACK in IPs) + (FC 9)
FC 14	Result of (RST out IPs) − 1
FC 15	Result of (RST in IPs) − 1

Sonification

In this stage, SoNSTAR assigns events using the extracted features and then assigns recorded sounds to them. Here, event conditions can be modified, new events can be created, and threshold values can be changed. Most of the interaction of the operator with SoNSTAR to explore and construct new events and to explore malicious behaviours and botnet patterns occurs at this stage.

SoNSTAR allows human interaction where the listener can interact with SoNSTAR in real time, and can choose to listen to part of the event sounds or to specifically targeted events representing certain behaviours and to ignore others or to change event conditions and assigned sounds.

Sound mapping and design

Sounds are mapped by assigning recorded sounds to events. Most of the events in the previous design have been mapped to the same original sounds with some modifications. A few events were dropped from the original design. In addition, new events are assigned to new recorded sounds. These recorded sounds are set to allow the user to recognise botnet events. SoNSTAR events are mapped to represent the occurrence of events derived from the SoNSTAR features. Table 6.1 illustrates the new feature-to-sound mappings.

At the end of a time window, algorithm A1 checks whether any local source IP (Table 6.2, row 1) has created 200 or more flows, and this feature is called "Src addr1 A count". It also checks whether any packet count (Table 6.2, row 3) is less than 15 packets and is repeated 250 times during the just completed time window. It counts the number of identical packet counts which have less than 15 packets passed through the destination ports. This feature is called "Identical packet counts1". Algorithm A1 is shown in Algorithm 2

At the end of the time window, algorithm A2 checks whether any local destination IP (Table 6.3, row 1) has created 200 or more flows, and the generated feature called "Src addr1 A count". It also checks whether any packet count (Table 6.3, row 3) is less than 15 packets and repeated 250 times during the just completed time window. It counts the number of identical packet counts which have less than 15 packets passed through the destinations ports. This feature is called "Identical packet counts2". Algorithm A2 is shown in Algorithm 3

At the end of the time window, algorithm A3 check each local destination host (Table 6.4, row 1) in array 3, for when the count of source hosts (Table 6.4, row 3) is greater than 84 and the count of destination ports (Table 6.4, row 5) is greater than 95 during the just completed time window. Algorithm A3 is shown in Algorithm 4

At the end of the time window, algorithm A4 checks each local source host (Table 6.5, row 1) in array 4, for when the count of destination hosts (Table 6.5, row 3) is greater than 84 and the count of destination ports (Table 6.5, row 5) is greater than 95 during the just ended time window. Algorithm A4 is shown in Algorithm 5

The thresholds for the above features were determined heuristically according to the nature, purpose, and expected traffic volumes for the specific network in question. Obviously, these would need to be adjusted for each separate network environment, though the above would serve as useful defaults for a network with modest numbers of visitors.

SoNSTARMain algorithm

The extended SoNSTAR system algorithm is shown in Algorithm 6.

Algorithm 2 Algorithm A1: Process the features of Array 1

```
Time-window period ended
Call Function with Array1 and index1
Open logs text file1 to write
for pointer1 <= array index do                    ▷ Get all rows information in the list
    Get local source address (Source IP) Column1 of Array1
    Get Sent packet count (Packet count) Column3 of Array1
    Rest (Src Addr1 Count1) = 0 and (Identical Packets Count1) = 0
    State1= False
    for pointer2 <= array index do                ▷ Compare with all sources list
        Get next source IP in the array (Src ip)
        Get next packet count in the array (P count)
        if SourceIP == source ip then
            increase (Src Addr1 Count1) by 1
            if SrcAddr1Count1 >= 50 then
                State1= True
            end if
            if SrcAddr1Count1 >= 200 then
                Write to logs file1, Anomaly
                Send message of Event 24 to Max/MSP Patch
            end if
        end if
        if Packet count == Pcount, And Packet count =< 15 then
            increase (Identical Packets Count1) by 1
            if State1 == True, And IdenticalPacketsCount1 >= 250 then
                Write to logs file1, Anomaly
                Send message of Event 25 to Max/MSP Patch
            end if
        end if
    end for
end for
```

Algorithm 3 Algorithm A2: Process the features of Array 2

```
Time-window period ended
Call Function with Array2 and index 2
Open logs text file2 to write
for pointer1 <= array index do                          ▷ Get all rows information in the list
    Get destination address ( Dest IP) Column1 of Array2
    Get total received packet count (Packet Count) Column3 of Array2
    Rest (Dst addr2 count2) =0 and (Identical Packets Count2) =0
    for pointer2 <= array index do                      ▷ compare with all dest IP's
        Get next destination IP in the array (dest ip)
        Get next packet count received in the array (p count)
        if Dest IP == dest ip then
            increase (Dst addr2 count2) by 1
            if Dst addr2 count2 >= 50 then
                State1= True
            end if
            if Dst addr2 count2 >= 200 then
                Write to logs file2, Anomaly
                Send message of Event 26 to Max/MSP Patch
            end if
        end if
        if Packet Count == p count, And packet count < 15 then
            increase (Identical Packets count2) by 1
            if State1 == True, And Identical Packets Count2 >= 250 then
                Write to logs file2, Anomaly
                Send message of Event 27 to Max/MSP Patch
            end if
        end if
    end for
end for
```

Algorithm 4 Algorithm A3: Process the features of Array 3

```
Time-window period ended
Call Function with Array3 and index 3
Open logs text file3 to write
Rest array C to collect malicious IP address list
for pointer <= array index do                           ▷ Get all rows information in the list
    Get destination address (dest IP)
    Get sources list and their count
    Get destination ports and their count
    if Src addr count3 >= 85, and Dst port count3 >= 95 then
        Write to logs file3
        Write to logs file3, Anomaly
        Send message of Event 28 to Max/MSP Patch
    end if
end for
```

Algorithm 5 Algorithm A4: Process the features of Array 4

```
Time-window period ended
Call Function with Array4 and index 4
Open logs text file4 to write
Rest array D to collect malicious IP address list
for pointer <= array index do                    ▷ Get all rows information in the list
    Get sources address
    Get destination list and their count
    Get destination ports and their count
    if Dst addr count4 >= 85, and Dst port count4 >= 95 then
        Write to logs file4, array extracted information
        Write to logs file4, Anomaly
        Send message of Event 29 to Max/MSP Patch
    end if
end for
```

6.3 Experimental work

6.3.1 Dataset

A number of traffic datasets are available to researchers. The 1999 DARPA Intrusion Detection Evaluation dataset contains three weeks-worth of traffic data.[1] The first and third weeks are attack-free, with the second week containing a variety of labelled attack traffic. However, it is not labelled with sufficient detail to support tool evaluation and does not indicate which are the malicious packets [24]. Furthermore, because some traffic has been inserted post hoc into the original data the dataset does not maintain trace consistency. The KD-99 dataset is based on the DARPA set and inherits its limitations. The CAIDA [26], PREDICT [129], and DEFCON [149] datasets all contain anomalous traffic but are not labelled. The University of New Brunswick provides several datasets of network traffic [154]. Their ISCX 2014 dataset is comprehensively labelled. However, the ISOT dataset from the University of Victoria [155] is labelled packet-by-packet and distinguishes normal from malicious traffic and so is well suited to the purposes of this study.

Evaluation dataset

The ISOT evaluation dataset [131] was used for the experiment. The dataset is an 11.39 GiB PCAP-format file and contains traffic conforming to the TCP, UDP, DNS and ICMP protocols. The ISOT dataset contains Strom, Waledac, and Zeus botnet command and control traffic. The dataset is labelled and contains a number of malicious and non-malicious flows

[1] https://ll.mit.edu/ideval/data/1999data.html.

Algorithm 6 SoNSTAR's main algorithm

```
Set Time-window period
Sniff packet and Get start time
if Packet == arrived then
    Unpack ethernet header
    Extract protocol
    if protocol == 8 then                                          ▷ IP packet
        Unpack IP header
        Extract source and destination addresses
        Extract transmission protocol
    else
        Get next packet from the sniffer
    end if
    if protocol == 6 then                                          ▷ TCP packet
        Unpack TCP header
        Collect and Extract (array Traffic flow, array IP flow) features
        Collect and Extract (array1, array2, array3 and array4) features
        Count IP flows and Traffic flows
        if Time − windowperiod == finished then
            Extract new features from Features Combiner
            Process Events of (array1, array2, array3 and array4) features
            Process Events of (array Traffic flow, array IP flow) features
            Write logs files while processing event conditions
            Send Event messages of to Max/MSP for sonification
        end if
        Get next packet from the sniffer a new Time-window 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
```

as shown in Table 6.7. The dataset was created from malicious traffic gathered by the French chapter of the Honeynet Project.

Table 6.7 Breakdown of ISOT malicious and non-malicious flows

State	No of flows
Malicious	55,904 (3.33%)
Non-malicious	1,619,520 (96.66%)
Total	1,675,424 (100%)

6.3.2 Experiment 1: Exploring traffic for botnet activity

Botnets mostly use distributed normal behaviour and stealth horizontal and vertical activity to prevent detection, but repetitive traffic patterns are still performed as bots are programmed to repeat scheduled tasks. For example, the botmaster sends an order through the C&C server to make its army of bots perform attacks against a specific web server. Since bots are usually high in number and use identical software, parts of their communication patterns will be identical or quite similar. Therefore, the victim web server is going to receive similar communication patterns from different bots, which is extremely unlikely in normal traffic.

In this work, no machine learning process is used but rather human-set events allow the operator to target different network behaviours. A single botnet attack will typically comprise several specific events. The operator recognises several event sounds in a different sequence. The sound type and sequence allow the operator to understand what is going on in their network. For example, in performing a SYN scan an adversary sends SYN packets to several ports on the target machine. SoNSTAR will play the sound of rain telling the operator that there are many SYN packets incoming to a specific host. Every open port then replies with a SYN-ACK packet, but closed ports reply with RST packets. Therefore, the target machine will send an RST packet against each closed port causing SoNSTAR to play a wind sound. So, when SoNSTAR generates rain followed by wind sounds, the operator would know that a SYN packet scan is happening. To use SoNSTAR to explore this traffic, the operator carries out the following steps.

1. Set appropriate time window period.
2. Run SoNSTAR to read from the ISOT dataset.
3. Listen to the sounds generated by SoNSTAR looking for any sounds indicating malicious behaviour or suspicious activity.

4. When a candidate sound is heard, open the log file corresponding to the event which triggered the sound.

5. Search the log file for a message indicating the local IP address and the time window number which caused the sound in order to confirm recognised behaviour.

6. Open the IP flow log file and look for the same time window number and then locate the local IP address obtained in the previous step.

7. Assign a recorded sound to the identified suspicious botnet pattern and set an event condition if the pattern is repeated twice in a time window period in order to confirm it is a botnet performing repetitive specific activities using identically programmed bots.

An initial time window was set at 35 seconds and then SoNSTAR was run on the dataset and its soundscape listened to. The repeated sequence of a bee colony sound (Table 6.1, row 26) followed by multiple rat squeak sounds (row 27) were observed during a single time window. The bee colony sound indicates that a local host has received connections from more than 200 different external hosts during this time window. This indicates a high possibility of distributed flow behaviour. The multiple rat squeak sound indicates that a local host has received the same number of packets from 250 different external hosts. This indicates very high possibility of repetitive flow behaviour. Since this behaviour is suspicious and also strange to be occurring in a single time window, it is suspected that this behaviour belongs to a botnet. Therefore, the log files are inspected to confirm how it has happened. Table. 6.8 shows part of the log file at time window 4 which contains these events.

It is observed that the local host `172.16.0.12` has received the same number of packets (13) from different external hosts into different ports. The log file also shows these port numbers are sequential. Also, the local host has connections with 497 different external hosts in this time window. All of this information leads us to suspect that this is botnet behaviour. Furthermore, the spring peeper sound (Table 6.1, row 28) indicates that a local host has received connections from more than 85 different external hosts through more than 95 different ports. This is suspected to be a horizontal scan but this depends on the purpose of the local host and its expected traffic. Also, it indicates that the local host might be part of a botnet communication network. Therefore, we looked into the log file. For example, at time window 8, the local destination host `172.16.0.11` has received connections from 461 different external hosts through 615 different ports, as shown in Fig. 6.5. Fig. 6.5 also shows part of the external host list. Fig. 6.6 shows part of the local port list at the local destination host. The complete log files can be found in the '`examples/logs`' folder on the SoNSTAR repository [41].

Table 6.8 Sample of vertical activity to local destination IP log file

Time window	Flow i.d.	Local Dest.	Local port	No. packets
4	483	172.16.0.11	2490	13
4	484	172.16.0.11	2491	13
4	485	172.16.0.11	2492	13
4	486	172.16.0.11	2507	13
4	487	172.16.0.11	2508	13
4	488	172.16.0.11	2509	13
4	489	172.16.0.11	2512	13
4	490	172.16.0.11	2513	13
4	491	172.16.0.11	2520	13
4	492	172.16.0.11	2521	13
4	493	172.16.0.11	2526	13
4	494	172.16.0.11	2527	13
4	495	172.16.0.11	2529	13
4	496	172.16.0.11	2531	13
4	497	172.16.0.11	2532	13

```
8 Dest IP: 172.16.0.11 Sources count: 461 Ports count: 615
8 Sources list: ['74.205.83.92' '208.56.131.207' '128.227.74.56' '203.84.221.51'
 '66.221.154.185' '70.84.121.39' '68.249.145.10' '209.191.88.239'
 '66.218.67.35' '67.63.20.218' '207.213.175.225' '72.249.26.99'
 '170.94.248.237' '216.84.45.242' '165.190.1.131' '128.115.249.90'
 '67.69.240.22' '12.3.33.11' '207.115.20.21' '216.39.53.1' '83.175.213.162'
 '213.81.152.19' '65.109.124.21' '66.218.66.70' '72.14.215.27'
 '64.79.170.114' '65.54.244.40' '70.158.51.118' '209.86.93.229'
 '65.119.39.207' '24.75.46.254' '207.28.212.5' '195.50.106.7' '65.54.245.8'
```

Fig. 6.5 Sample of external horizontal scan log file, part 1

```
'66.170.2.43' '213.199.154.22' '212.115.192.194'] ports list: ['10004'
 '10005' '10006' '10007' '10012' '10018' '10019' '10020' '10021'
 '10029' '10030' '10031' '10034' '10039' '10044' '10048' '10049' '10050'
 '10052' '10053' '10060' '10066' '10067' '10070' '10073' '10077' '10080'
 '10083' '10084' '10085' '10091' '10095' '10096' '10101' '10102' '10105'
 '10110' '10112' '10113' '10120' '10123' '10126' '10127' '10129' '10131'
 '10133' '10136' '10138' '10142' '10147' '10150' '10153' '10156' '10159'
 '10161' '10162' '10165' '10171' '10178' '10181' '10182' '10183' '10184'
```

Fig. 6.6 Sample of external horizontal scan log file, part 2

Result and botnet patterns

We identified two local IP addresses `172.16.0.11` and `172.16.0.12` which exhibit suspect behaviour over several time widows. Based on our knowledge of botnet characteristics, this confirms that botnet behaviour is detected.

One of the purposes of the experiment was to determine whether or not SoNSTAR features can create events capable of targeting specific behaviour. Clearly, SoNSTAR features and events can be used to draw a normal traffic base line for traffic behaviour.

One of the purposes of SoNSTAR is to support existing IDSs and to contribute to their development by helping human operators to discover features, events, and patterns that can be used by IDSs to detect malicious behaviour. Therefore, to use SoNSTAR to detect and confirm botnet patterns, the operator carries out the following steps.

1. Open the IP flow log file and look for the same time window number and then locate the local IP address obtained in the previous steps.

2. Assign a new recorded sound to the suspected botnet pattern and set an event condition to play that sound when the pattern occurs twice in a time window. The repetition of this event indicates that bots are performing repetitive specific flow patterns.

We opened the IP flow log file and obtained IP flow patterns associated with the suspicious traffic. Feature conditions to match the traffic were constructed and mapped to sounds as shown in Table 6.7. Then, when SoNSTAR was run, the sounds of a squirrel running quickly and a rat moving in dry leaves were heard which confirmed the presence of botnet behaviour. Therefore, IP flow features demonstrated very advanced capabilities to construct patterns which can be used to detect botnet traffic. SoNSTAR could help the user to discover botnet behaviour by only mapping for repetitive normal patterns within the TCP traffic without the need to consider these attacks are happening at the application layer. The IP flow log files can be found in the '`examples/logs`' folder on the SoNSTAR repository [41].

Fig. 6.8 shows part of the IP flow log file indicating how some botnet patterns use normal behaviours that are repeated several times within a single time window. This represents part of time window number 14 from the IP flow numbers 19 to 30. The IP flow log file can be found in the repository.

Feature construction: Example 1

Event 2 in Table 6.1 has the following event condition based on the three way handshake mechanism, and the function of the RST packet in the TCP protocol: 'SYN in IPs >10 and SYN in IPs <30 and PSH ACK out IPs == 0 and RST out IPs > 0'

```
14&19&172.16.0.11&194.176.201.22&1&1&1&0&0&1&0&0&2&4&7&7&0&0&0\\
Anomaly. rebot activity A 111
14&20&172.16.0.11&64.250.228.130&1&1&1&0&0&1&0&0&2&4&7&7&0&0&0\\
Anomaly. rebot activity A 111
14&21&172.16.0.11&222.255.37.15&1&1&1&0&0&1&0&0&2&4&7&7&0&0&0\\
Anomaly. rebot activity A 111
14&22&172.16.0.11&216.200.145.235&1&1&1&0&0&1&0&0&2&4&7&7&0&0&0\\
Anomaly. rebot activity A 111
14&23&172.16.0.11&64.233.183.27&3&3&3&0&0&3&0&0&6&12&21&21&0&0&0\\
Anomaly. rebot activity A 111
14&24&172.16.0.11&207.115.21.22&1&1&1&0&0&1&0&0&2&4&7&7&0&0&0\\
Anomaly. rebot activity A 111
14&25&172.16.0.11&217.72.192.149&1&1&1&0&0&1&0&0&2&4&7&7&0&0&0\\
Anomaly. rebot activity A 111
14&26&172.16.0.11&13.8.138.217&1&1&1&0&0&1&0&0&2&4&7&7&0&0&0\\
Anomaly. rebot activity A 111
14&27&172.16.0.11&60.229.18.61&1&1&1&0&0&1&0&0&2&4&7&7&0&0&0\\
Anomaly. rebot activity A 111
14&28&172.16.0.11&12.168.122.203&1&1&1&0&0&1&0&0&2&4&7&7&0&0&0\\
Anomaly. rebot activity A 111
14&29&172.16.0.11&64.5.42.8&1&1&1&0&0&1&0&0&2&4&7&7&0&0&0\\
Anomaly. rebot activity A 111
14&30&172.16.0.11&144.140.80.13&1&1&1&0&0&1&0&0&2&4&7&7&0&0&0\\
Anomaly. rebot activity A 111
```

Fig. 6.7 Part of the IP Flow log file ('rebot' = repeated botnet).

This means that if a host received 10–30 SYN packets requesting a connection but returned no data (zero PSH-ACK packets and one or more RST packets) then the rain-on-roof sound should be played. This sound will tell the operator that the network is receiving requests for connection on multiple ports but no data was returned. This is analogous to customers repeatedly coming into a shop, asking the price of an item, and then leaving without making a purchase. As any closed port will sent an RST packet on receipt of any incoming packet type, this behaviour is indicative of a port scan.

Feature construction: Example 2

We will use same example of time window 8 where we recognised the IP address 172.16.0.11. So, we open the IP flow log file and search for time window 8 and the IP address 172.16.0.11. An extract of this time window log is shown in Table 6.9.

Using this information we can build a pattern. The first IP flow we see is flow 412 between hosts 172.16.0.11 and 195.188.53.99. We can see the number of FIN out and FIN in, SYN out, and SYN-ACK in packets are greater than 0 and are all equal. Therefore, the first part of the pattern is the condition: 'FIN out IPs == FIN in IPs == SYN out IPs == SYN ACK in IPs > 0'.

We know from the TCP protocol that following a FIN in and SYN out pair, an ACK out packet is a confirmation of the communication. We see that in IP flow 412, ACK out = FIN

Table 6.9 Sample of vertical activity to local destination IP log file

Time window	Flow	Host A	Host B	Packet counts														
				FIN		SYN		SYN-ACK		RST		ACK		PSH-ACK		URG-PSH-FIN	NULL	LAND
				Out	In	Out	In	Out	In	Out	In	Out	In	Out	In			
8	412	172.16.0.11	195.188.53.99	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	413	172.16.0.11	82.185.226.116	2	2	2	0	0	2	0	0	4	8	14	14	0	0	0
8	414	172.16.0.11	66.193.69.2	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	415	172.16.0.11	63.166.215.100	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	416	172.16.0.11	217.116.0.152	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	417	172.16.0.11	80.240.225.37	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	418	172.16.0.11	195.242.120.10	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	419	172.16.0.11	209.59.136.109	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	420	172.16.0.11	80.12.242.15	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	421	172.16.0.11	66.218.66.215	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	422	172.16.0.11	205.152.58.32	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	423	172.16.0.11	143.100.37.72	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	424	172.16.0.11	213.161.248.130	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	425	172.16.0.11	212.88.148.234	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	426	172.16.0.11	64.251.84.10	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	427	172.16.0.11	80.207.150.20	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0
8	428	172.16.0.11	66.216.121.101	1	1	1	0	0	1	0	0	2	4	7	7	0	0	0

in + SYN out, so we add another condition to the pattern: ‘ACK out IPs == FC9 where FC9 = FIN in IPs + SYN out IPs’ (see Table 6.6).

We also see that the number of ACK-in packets is greater than ACK-out by 2, so the pattern is extended by the condition ‘FC10 >= 2 where FC10 = ACK in IPs - ACK out IPs’.

Next we observe that PSH-ACK-out and PSH-ACK-in packet counts are equal (7), so the condition ‘PSH ACK out IPs == PSH ACk in IPs >= 1’ is added.

The condition ‘ACK in IPs < PSH ACk in IPs’ is added to reflect the relationship between the number of ACK-in and PSH-ACK packets.

Therefore the complete pattern is ‘FINoutIPs == FINinIPs == SYNoutIPs == SYNACK-inIPs >0 and ACKoutIPs == FC9 and FC10 >1 and PSH ACK out IPs == PSH ACK in IPs > 0 and ACKinIPs < PSH ACK in IPs’, which is event condition 30 in Table 6.10. If this pattern is repeated twice in a time window then this is indicative of botnet activity. Because normal traffic takes the form of a random pattern it cannot be repeated as quickly as it happens in the case of botnet networks especially if it repeats in a single time period several times.

Table 6.10 Feature-to-Sound Mappings of Botnet Patterns. The squirrel sound is used when multiple sources repeatedly target a single host. The rat sound denotes a single host receiving the same number of packets across multiple ports.

No	Feature Conditions	Sound
30	FIN out IPs == FIN in IPs == SYN out IPs == SYN ACK in IPs >0 and ACK out IPs == FC 9 and FC 10 >1 and PSH ACK out IPs == PSH ACK in IPs >0 and ACK in IPs <PSH ACK in IPs (if repeated twice in the same time window (if R2T))	Squirrel running quickly
31	FIN out IPs == 0 and FIN in IPs == SYN out IPs == SYN ACK in IPs >0 and ((ACK out IPs == FC 9 and RST out IPs ==0)or (ACK out IPs == FC 9 +1 and RST out IPs == 0)or (ACK out IPs == FC 13 and RST out IPs ==1)) and ACK in IPs <ACK out IPs and PSH ACK out IPs >PSH ACk in IPs >0 and RST in IPs == 0 (if R2T)	Rat moving in dried leaves
32	FIN out IPs == FIN in IPs == SYN out IPs == SYN ACK in IPs == ACK in IPs >0 and RST in IPs == FC 9 and ACK out IPs == PSH ACk out IPs == PSH ACk in IPs >0 and ACK in IPs <ACK out IPs and SYN in IPs == SYN ACK out IPs == RST out IPs == 0 (if R2T)	Squirrel running quickly
33	FIN out IPs == FIN in IPs >0 and SYN out IPs == SYN ACK in IPs == ACK in IPs >0 and RST in IPs == FC 11 and ((ACK out IPs == PSH ACk in IPs >0 and (PSH ACk out IPs == FC 12 or PSH ACk out IPs == PSH ACk in IPs)) or (PSH ACk in IPs == PSH ACk out IPs >0 and ACK out IPs == FC 12 or ACK out IPs == PSH ACk in IPs)) and ACK in IPs <ACK out IPs and SYN in IPs == SYN ACK out IPs == RST out IPs == 0 (if R2T)	Squirrel running quickly
34	FIN out IPs == 0 and FIN in IPs == 0 and SYN out IPs == 1 and SYN in IPs == 0 and SYN ACK out IPs == 0 and SYN ACK in IPs == 1 and RST in IPs == RST out IPs == 0 and ACK out IPs == 1 and ACK in IPs == 0 and PSH ACk out IPs == 0 and PSH ACK in IPs == 0 (if R2T)	Rat moving in dried leaves

Continued on next page

Table 6.10 – *Continued from previous page*

No	Feature Conditions	Sound
35	FIN out IPs == FIN in IPs >0 and SYN out IPs == SYN in IPs == SYN ACK out IPs == SYN ACK in IPs == RST out IPs == 0 and RST in IPs == FC 11 and ACK out IPs == PSH ACK out IPs == PSH ACk in IPs >FC 11 and ACK in IPs <ACK out IPs (if R2T)	Squirrel running quickly

6.3.3 Experiment 2: Using SoNSTAR as an IDS to validate discovered patterns

Acting as network operators, we used SoNSTAR to identify the behaviour of botnets inside the ISOT dataset traffic. As a result, through sound and log files, we were able to discover six IP flow patterns which indicate bot behaviour. We consider using SoNSTAR as a passive IDS based on the patterns discovered. Therefore, we added a detection algorithm to detect botnets in the dataset based on the IP flow patterns shown in Table 6.10.

The results need to be evaluated based on a labelled dataset and compared against other methods using the same dataset. Therefore, we had to configure the detection algorithm for every time window to classify and label each detected flow as normal or malicious before storing the results in a log file. Table 6.11 shows how the resulting log file is structured.

The log file columns are, from left to right, the time window, the flow number in the time window, the flow number in the dataset, host A, host B, SoNSTAR classification result, and the label derived from the labelled dataset.

Evaluation metrics

To evaluate SoNSTAR's anomaly detection classifier based on IP flow patterns, performance was assessed based on the following metrics:

1. The number of true positives (TP) where SoNSTAR correctly classifies a malicious flow.

2. The number of true negatives (TN) where SoNSTAR correctly classifies a normal (non-malicious) flow.

Table 6.11 SoNSTAR classification log file excerpt. The SoNSTAR classifications can be compared against the labels in the ISOT dataset.

Time window	Time window flow id	Dataset flow i.d.	Host A	Host B	Classification	
					SoNSTAR	ISOT
75	42	11174	172.16.0.12	65.54.244.40	Malicious	Malicious
75	43	11175	172.16.0.12	65.55.88.22	Malicious	Malicious
75	44	11176	172.16.0.12	216.39.53.3	Malicious	Malicious
75	45	11177	172.16.0.12	128.192.1.108	Malicious	Malicious
75	46	11178	172.16.0.12	65.54.245.72	Malicious	Malicious
75	47	11179	172.16.0.12	65.54.245.8	Malicious	Malicious
75	48	11180	172.16.2.13	203.69.42.35	Normal	Normal
75	49	11181	172.16.2.2	69.147.121.161	Normal	Normal
75	50	11182	172.16.2.12	203.84.202.164	Normal	Normal
75	51	11183	172.16.2.13	87.248.113.14	Normal	Normal
75	52	11184	172.16.2.2	209.85.135.103	Normal	Normal
75	53	11185	172.16.2.2	209.85.135.147	Normal	Normal

3. The number of false positives (FP) where SoNSTAR mistakenly classifies a normal flow as malicious (botnet) activity.

4. The number of false negatives (FN) where SoNSTAR mistakenly classifies a malicious flow as a normal flow.

Results

The SoNSTAR detection algorithm was run three times to read the whole ISOT dataset with three different time windows (20, 30, and 60 s) to evaluate the reliability of the discovered IP flow patterns. Table 6.12 shows the SoNSTAR as an IDS detection results for the three different time windows. Table 6.13 show precision, recall, F-measure, accuracy and the false positive rate (FPR) results achieved for the three time windows..

Table 6.12 SoNSTAR as an IDS detection results.

Time window	Results			
	FP	FN	TP	TN
20 s	152	144	12,543	510,720
40 s	136	75	11,105	463,000
60 s	312	49	10,632	443,819

Fig. 6.8 shows how the TPR/recall increases with time window size.

Table 6.13 Comparison measures with time window of 20 s, 40 s and 60 s.

Time window	Results				
	Precision	**TPR/recall**	**F-measure**	**Accuracy**	**FPR**
20 s	98.8	98.86	98.83	99.94	0.0297
40 s	98.7	99.32	99.05	99.95	0.0293
60 s	97.14	99.54	98.33	99.92	0.07

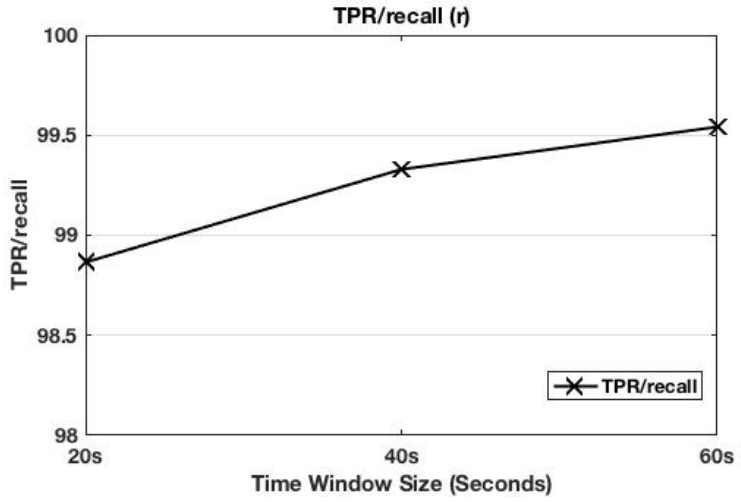

Fig. 6.8 TPR/recall

Comparison

The SoNSTAR experiment used the same ISOT evaluation dataset and metrics as Kirubavathi and Anitha [92]. They evaluated three different classifier techniques; namely, the Boosted decision tree ensemble classifier (AdaBoostM1+J48), Naive Bayesian (NB) statistical classifier and the Support Vector Machine (SVM) discriminative classifier. Kirubavathi and Anitha used the whole ISOT data set for testing, having used other data for training. Their results are based on three time windows of 60 s, 120 s, and 180 s. SoNSTAR was not studied at 120 s and 180 s because it is designed to represent network traffic as close as possible to real-time. Such long time windows would mean the user having to wait for two or three minutes before any change in state can be heard. Table 6.14 compares the SoNSTAR results achieved above with the three systems measured by Kirubavathi and Anitha with a 60 s time window. The comparison result shows that the six patterns discovered by the operator through sonification achieves a better detection accuracy and precision and very low false positive rate. We conclude that involving human understanding in the detection of botnets increases the chances of detecting them, especially when detecting new botnets.

Table 6.14 Comparison with existing method at time window of 60 s.

Classifier	Results				
	Precision	TPR/recall	F-measure	Accuracy	FPR
AdaBoostM1+J4 [92]	0.958	0.933	0.954	94.13	0.12
NB [92]	0.979	0.976	0.968	95.86	0.04
SVM [92]	0.936	0.943	0.939	91.37	0.14
SoNSTAR	0.971	0.995	0.983	99.92	0.0007

6.4 Discussion

Botnets are increasingly becoming a major threat to organisations, communication infrastructures and economies. Botnet activity is usually stealthy and hard to detect automatically which means that the human operator plays a vital role in identifying its presence. SoNSTAR enables its user to monitor and discover behaviours which may be part of a botnet attack. The results achieved in this chapter show a high potential for removing barriers to administrators' understanding of their networks' activity.

Table 6.8 shows how a local destination host (`172.16.0.11`) has received the same number of packets (13) on different flows from several hosts. This indicates pre-programmed behaviour implementing the same action over sequenced port numbers. Fig. 6.5 and Fig. 6.6 show a sample of the log file which the operator can use to gain more understanding of the situation after hearing sounds that indicate potential botnet behaviour. The sudden increase in the number of the external hosts (461) connecting to the same local-host and the number and range of port numbers provide deeper explanation and understanding of the state of the traffic. This example supports the integration between sonification and visualisation approaches.

Fig. 6.7 shows how the local host receives the same communication procedures from several external hosts. Although each individual flow considered as normal behaviour, the correlation between these flows shows repetition of procedures which indicate the presence of a pre-programmed Botnet generating this traffic.

Table 6.12 shows the TPR was highest in the 60 s time window (99.54%). Fig. 6.8 shows that the accuracy increases with increasing time window duration. This indicates that the longer the time window the more likely an operator is to notice botnet flows. Table 6.12 illustrates SoNSTAR achieved excellent precision, recall, F-measure, accuracy, FPR results during all time periods. In particular, the FPR is low, which confirms that the involvement of human understanding in the selection of events practically affects the reduction of the FPR.

These results confirm that sonification could have a significant impact on the development of features and events required for the development of protection systems.

The repetition of IP flows patterns of the Botnet traffic confirms that even IP flows patterns of normal behaviour traffic when repeated in the same time window would be considered suspicious. The use of IP flow patterns open high potential to learn more about network traffic and malicious behaviours.

The use of sonification and log files to monitor network traffic allows the user to explore, learn and transfer their understanding directly to event creation. This allows them to experiment with many event possibilities which can be included to detect the targeted behaviour.

SoNSTAR enables the user to explore distributed, parallel and horizontal behaviours that are similar to normal behaviours but which can be linked to DDoS or botnet characteristics. Although every single flow may appear normal in and of itself, the overall behaviour is suspicious. Moreover, the user recognises identical IP flow patterns repeated several times within the same time window and probably also within several subsequent time windows. It is not expected to see such behaviour within normal traffic. It is as if a number of website visitors increased suddenly and performed a similar action. From a machine this may look like normal behaviour, but for the human mind it is suspicious and does not look right. Therefore, normal behaviour from several different external hosts cannot use a specific communication mechanism unless it was programmed to perform such action.

Acting in the role of network operator, we have discovered some repeated identical behaviours within traffic flows. In our study of these flow patterns, we found that it would be impossible for this normal behaviour to be repeated several times in a single time period, which indicates that it is botnet behaviour. To facilitate the detection of the discovered patterns in the future, we have mapped them into recorded sounds.

The advantage of using sonification is that the user is informed immediately about any traffic activity and the sounds are easier to follow and comprehend. Visualisation is capable of representing traffic behaviour, but it is difficult for the user to follow and recognise the frequency and sequence of occurrence of events in the way that sonification can provide, not least because it would require constant attention to the visual display. Therefore, sonification and visualisation have to be integrated to raise security situational awareness.

Any features, events or patterns discovered to be symptoms of malicious behaviour could then be passed to any IDS and tested and used in a machine learning process afterwards. SoNSTAR improves situational awareness levels and allows users to learn more about their own network environment rather than studying network behaviour in general. As soon as the SoNSTAR user starts monitoring the network, they will recognise various thresholds and

normal behaviours that pertain to their network environment and after some time will be easily able to distinguish different normal behaviours. Any unusual 'normal' behaviour will become a suspicious behaviour which will improve the user's awareness level.

The operator can use SoNSTAR to study the vulnerability of a network. For example, the operator could perform a penetration test against the network while monitoring it with SoNSTAR. The user can perform expected attacks based on the network's purpose and any adversary's motivations. SoNSTAR will be able to help the user to create events which reveal those attacks even if they take the form of normal behaviour. It is provide

This work deals with the extension of SoNSTAR by mapping TCP traffic flow features to sound such that it enables the human operator to recognise botnet activity and patterns without the need to manually inspect the traffic's content. The first contribution is the new features extracted that target parallel, horizontal, distributed and repetitive flow behaviours plus four new algorithms to process event feature conditions.

The second contribution is the discovery and definition of six patterns of botnet behaviour based on IP flow. The significance of this discovery is that botnets exhibit unique repetitive IP flow patterns which can be used to detect them at the network layer instead of the growing demand to detect botnets at the application layer. The third contribution consists of using IP flow patterns for classification instead of using packet patterns, which opens up a new path of research to find better ways to develop IDSs in the future based on IP flows. Finally, as a fourth contribution, the proposed sonification tool (SoNSTAR) is an interactive, flexible, and scalable approach for botnet traffic detection that can be adjusted according to the understanding of the human operator and future security demands, and this is the first sonification solution to target botnet detection.

Further development can be conducted to represent log file information in a visual manner, which would enable more real-time integration between sonification and visualisation. Furthermore, four new algorithms were added to the proof-of-concept SoNSTAR system and no detrimental impact on the system performance was observed. However, it would be instructive to run performance tests to determine the scaleability of adding successive algorithms for dealing with new traffic features. SoNSTAR can be installed on a network gateway or, if the network is very large with lots of traffic, then multiple instances could be installed on subnet gateways. Further work is needed to determine the thresholds for making such decisions.

6.5 Summary

IDS technologies do not include the protocol flow granularity required to understand network events inside an environment. Our proposed solution is to use SoNSTAR to learn about those environments and then IDS technology could be developed specifically for specific environments and can be deployed with confidence in detecting malicious activity. This chapter describes the evaluation of the events features of SoNSTAR that address the specific environment, including its coordinated sonification and report building capabilities. Also, it illustrates how it can be used to discover the unexpected behaviours in network flow data in labelled datasets with different attacks and normal and malicious behaviours.

This chapter described a novel and innovative method to tackle botnet issues. This represents the first mechanism for sonifying botnet behaviour. Its objective is to target botnet events in order to enable the operator to recognise them. To successfully achieve this target, we have introduced new extracted features that create events which can target botnet behaviour. SoNSTAR does not use any botnet detection algorithm, but enables a human operator to recognise botnets by linking sounds of events to the structure of botnet behaviour and then to extract botnet IP flow patterns, and then to confirm the presence of botnet activity by finding those patterns repeated within a time window.

We defined six patterns of botnet behaviour representing normal flow patterns used by botnets. We found evidence of botnets having a unique repetitive IP flow patterns from the ISOT dataset. This shows that our sonification approach and IP flow structures can be used to detect known botnets as well as novel ones. And we have demonstrated in experiments that our sonification mechanism is effective in revealing important aspects of botnet patterns. The pattern validation experiment shows how patterns discovered by SoNSTAR can be used by IDSs to prevent a zero-day attack.

Based on this an experiment was performed to target and detect botnet behaviours in computer network. This chapter describes the traffic dataset used in the experiment. In addition, it describes the experimental design, procedure, and results.

Further research and development can be conducted to represent log file information in a visual manner, which would enable more real-time integration between sonification and visualisation.

Chapter 7

Conclusions and Future Work

This chapter presents the conclusions of this thesis along with a brief description of its contributions and some directions for future work. The main conclusions of the research are presented in Section 7.1 and a summary of contributions is given in Section 7.2. Section 7.3 outlines the challenges faced and the solutions offered, whereas Section 7.4 highlights the limitations of the proposed approach. The directions for future research are then discussed in Section 7.5.

7.1 Thesis summary

This research addresses the question of "how can sonification be used in maintenance of real-time situational awareness to provide the protocol flow granularity required to understand the network environment behaviour?" This thesis has presented research on real-time sonification of computer network traffic to support cyber security situational awareness as part of measures that could be implemented to enhance the existing security tools portfolio. Chapter 2 presented a literature review on computer networks security and situational awareness and provided the relevant background of the existing security tools, traffic protocols and malicious behaviours. Chapter 3 presented a literature review on sonification in general and on existing sonification systems for computer network monitoring. In addition, Chapter 4 introduced and described our (SoNSTAR) system and explained the SoNSTAR design and the feature extraction and sound mapping techniques used. It also described the two primary experiments conducted, the first of which aimed to test the packet count concept, while the second targeted the evaluation of sonic detection. Chapter 5 proceeded by providing the details of an experiment conducted to evaluate the efficiency and effectiveness of SoNSTAR against the Snort IDS and has discussed the contributions of SoNSTAR towards raising the cyber security situational awareness levels. In addition, the experimental procedures, results

and evaluation, and further discussions were all detailed in the chapter. Chapter 6 presented new techniques to extract and map additional traffic features as part of a further development of SoNSTAR to target botnet behaviours, and described an experiment conducted to use of SoNSTAR as a passive IDS based on the discovered botnet patterns through utilising sonification. The results of the evaluation have demonstrated the efficacy of using this approach to deal with botnet traffic compared with other research based on the same datasets.

7.2 Contributions of this Thesis

This thesis makes three main contributions to the field. The first is the SoNSTAR system itself and the supporting evaluations. The second is the introduction of the concept of IP flow which, together with feature construction methods and techniques for representing multiple identical events with a single sound, reduces the complexity of network traffic such that it becomes possible to monitor all the traffic passing through the network. The third concerns the use of sonification in the discovery of malicious network behaviours, demonstrated in Chapter 6 with a specific case study dealing with botnet activity. These will now be expanded upon in turn.

7.2.1 SoNSTAR

SoNSTAR provides a real-time soundscape sonification monitoring system that does not require the user to be dedicated to watching a visual display screen. SoNSTAR uses a method that collects selective status information, from which it periodically extracts features to free memory storage and save processing power. Unlike previously reported network sonification systems, SoNSTAR presents the state of raw network traffic in real time in such a way that the resultant soundscape can be used for real monitoring. Other sonification systems (e.g., Peep [59] and NetSon [167]) did communicate aspects of traffic (such as information about IP addresses, traffic volumes and port numbers) but SoNSTAR is the first system to use detailed information from individual packets to allow knowledge about actual traffic behaviour to be constructed. The focus on flows rather than source and destination addresses allows malicious network activity to be detected and recognised by human operators.

In addition, this research is the first to evaluate network sonification with human participants and the studies described in Chapters 4 and 5 demonstrate that SoNSTAR's output is comprehended. The study in Chapter 5 revealed that using SoNSTAR for monitoring a computer network was shown to be more useful in supporting a human operator to identify suspicious network activity than a current leading intrusion detection system (Snort) and that

the use of sound reduced the overall mental workload. This is the first study of this kind to demonstrate the usefulness of network sonification for this type of monitoring activity.

Moreover, the study in Chapter 4 showed that the soundscape representation of traffic is better than MIDI messages due to the variety of natural and man-made sounds available and the advantages of sound comprehension. This study has indicated that using sonification improved the monitoring process, even for people who only have basic knowledge of network monitoring. The experiment in 4.6 has also demonstrated humans' impressive capability to recognise recorded sounds.

7.2.2 IP flow and feature construction

The complexity and volumes of modern network traffic militate against real-time sonification and visualisation of the entire traffic. It has not been possible to identify events as they are happening without significant delay or with explanations as to what exactly happened. IDSs and anomaly detection systems are good at blocking certain activity, but the explanation of the traffic activity requires the extensive post-incident review of log files.

IDSs use the concept of traffic flows to identify connections between two hosts. However, a great many traffic flows can exist between a single source and destination host due to the large number of IP ports that can be addressed, with each port connection constituting a single traffic flow. Therefore, since network traffic volumes can be huge, the second main contribution of this thesis is the introduction of the concept of the "IP flow" which aggregates all traffic flowing between two hosts, thereby allowing the traffic to be represented as a single IP flow, thus resulting in a considerable reduction of the amount of information required to be represented. The volumes of data are further reduced by inspecting the status flags of each packet within an IP flow and maintaining counts of each packet type. This major contribution results in reducing the complexity of huge volumes of traffic so that they can be sonified in a comprensible manner by using the IP flow to detect network behaviours, especially vertical behaviours.

Using IP flows (and the packet count aggregations within), further benefits are obtained. First, analysis of the counts of the various packet types allows distinctive features (and combinations of features) to be collected which identify different types of traffic behaviour. These features can then be joined using relational operators to define events which signal specific types of activity, enabling the recognition of normal, anomalous and malicious traffic.

To prevent overloading of processor resources, the SoNSTAR approach collects traffic features over a series of time windows, and at the end of each time window sending the event information for sonification and subsequently freeing memory storage. According to the author's knowledge, this contribution has not been achieved before in any other existing

sonification system. The technique has succeeded in reducing the number of sounds in each time window by representing similar flow events only once.

7.2.3 Sonification for discovery of malicious activity

SoNSTAR has been designed to provide a flexible sonification framework. The third contribution of this research is the implementation of a technique that allows the development of new features and events to target and detect repetitive, parallel, distributed and horizontal behaviours and to sonify those events in a human-comprehensible form. As operators learn more about their own network, they can extend the feature set to account for their own individual circumstances. This offers a flexibility that is not present in current network monitoring or intrusion detection systems. The very act of listening to the traffic generates a fast discovery process leading to new knowledge of malicious behaviours that is not possible with current algorithmic approaches. SoNSTAR's interactivity lets an operator explore their network in new ways and discover gaps in the network security. This knowledge can then be included in the IDS.

The case study in Chapter 6 showed how SoNSTAR can be used by an operator to construct new feature combinations and events to target the particular type of activity associated with botnets. This significant benefit supports intrusion detection systems and enhances their ability to detect zero-day attacks. The results of the case study have shown that our botnet patterns set discovered using SoNSTAR achieved better accuracy and recall rates in detecting botnet activity than other studies that used the same dataset. Moreover, this case study has demonstrated that using SoNSTAR based on discovered events and features of function-specific network traffic enables the discovery of network vulnerabilities based on the user's understanding of the motivation behind the cyber-attacks expected and studying the unique traffic behaviour of their network. Furthermore, this system shows that botnet attacks launched at the application layer to evade detection can be detected at the network layer when a human operator is in the loop, and this enables the operator to quickly gain an understanding of the situation.

7.2.4 Summary

The development of SoNSTAR changes the philosophy of relying solely on machines using specific algorithms and rules to detect malicious behaviour. This system opens the door to dynamic interaction with network traffic bringing the human into the loop by leveraging our intuition, reasoning, and pattern recognition abilities in the process of detecting malicious activity. This has not been demonstrated in any previous sonification monitoring systems.

SoNSTAR includes the protocol flow granularity required to understand network events inside a network environment, which is not provided by current IDS technologies and other monitoring tools. SoNSTAR users are capable of recognising novel behaviour changes in a very short time window. SoNSTAR allows its users to tune their network according to their own specific network environment in order to extract a unique normal baseline of their network behaviour. SoNSTAR's algorithms represent a model for innovators to develop and enhance sonification tools. Finally, SoNSTAR could be used in educational settings to enable students to learn about network environments and structures.

The sonification of network traffic based on flag state counts and network flow techniques in addition to soundscape mapping has enabled our solution to provide a valuable contribution to the fields of sonification, computer network security, and situational awareness. The experiment in 5.1 has clearly demonstrated improvements in monitoring when using sonification compared to the visual method only. Although the participants were not computer security professionals, they were able to use SoNSTAR after a very short training period. The results have also shown that monitoring based on sonification reduced the workload of users in comparison to the visualisation approach. Temporal demand, effort and frustration rates have all been shown to improve under sonification use.

Two main lessons were learned from the experiments. First, sounds and noises with similar characteristics should be avoided because participants them hard to differentiate. Second, to obtain more accurate results in terms of the workload of users, it would be necessary to increase the duration of the monitoring period so that the waiting period is sufficient to show the difference between the use of sonification and visualisation techniques in real monitoring conditions.

7.3 Challenges and solutions

Despite concerted efforts to reduce the impact of increased traffic in computer networks on using sonification to represent the state of traffic in real-time, current sonification techniques for security monitoring are inadequate. Intrusion attacks are becoming more stealthy and complex. Moreover, existing monitoring tools do not provide the protocol flow granularity and flow status representation techniques required to understand the computer network environment. Since the behaviour of attack traffic has become much like normal traffic behaviour, monitoring tools that depend on traffic volume behaviour changes are becoming increasingly unable to detect attacks. The involvement of the human mind in the process of detection has become increasingly important due to the presence of other factors that the detection process depends on, such as the nature of the network and its purpose and the motivation behind the expected attacks.

In addition to the technical difficulties faced during development of SoNSTAR, there were three main challenges involved. The first challenge involved the presence of a huge amount of traffic that needed to be represented using sonification in a way that allowed human understanding of the complete traffic status. To tackle this, we have introduced the IP flow which has resulted in reducing the number of flows required to be sonified to represent the entire traffic. The IP flow features have demonstrated extensive capabilities in creating flow events that enable the SoNSTAR user to target any traffic behaviour.

The second challenge was ensuring that the sonification approach is interactive and allows the user to recognise and comprehend sounds of events immediately. Since monitoring is a continuous process, the sounds have to be recognisable with the least sound fatigue. As a solution, we have used event-to-sound mappings to transform the network environment into a soundscape environment. Furthermore, we have used recorded sounds of nature and animals which humans already know, and used this prior knowledge of these sounds to facilitate the comprehension of the network behaviour. For example, the sound of fire in the woods would allow the user to understand the nature of the malicious behaviour. For interaction, we have enabled the user to create and map events to the sounds as they like. They can experiment with events thresholds and change each event sound's volume.

The third and final challenge involved collecting features to target repetitive, parallel, distributed, vertical, and horizontal behaviours, which are used by some attacks as evasion methods to avoid detection. As a solution, we have introduced four algorithms to address this challenge, and the features collected have subsequently enabled the detection of botnet behaviour.

7.4 Limitations

One of the advantages of sonification over visualisation techniques is that the latter require the operator to be attached to a monitoring screen for long hours, while in former, the user can perform other essential tasks while listening to the network. Since monitoring is a continuous process, the evaluation of sonification systems requires participants to perform the monitoring task for hours to measure the participants' performance and workload in real life monitoring conditions.

Since the behaviour of each network is different, we could not obtain different real network environments in which to use SoNSTAR. Therefore, we intended to evaluate the potential of drawing a baseline of normal traffic behaviour of different networks and to study their behaviour during a penetration test while monitoring the traffic using SoNSTAR. We consider this helps to raise cyber security situational awareness and to establish a baseline of normal traffic within the specified network.

7.5 Future research directions

This research and its outcomes have highlighted further work that could be done to extend SoNSTAR. Our real-time sonification system, with its important features and sound mapping design and implementation, helps to address the cyber security situational awareness objective. However, because the focus of this thesis was on the TCP protocol, further research needs to be conducted to investigate how sonification can be applied to the other network traffic protocols. Furthermore, integration with visualisation techniques could be applied to support the sonification solution.

There are various possible areas for future work that can be carried out to validate and further develop SoNSTAR, as a continuation to the research work presented in this thesis. These areas include the following:

1. Building on the developed SoNSTAR feature extractor and combiner. We succeeded in creating many features that can be submitted to publicly open databases for other researchers to use in developing their own detection systems. Future work could involve developing a language for describing these discovered features and events based on network traffic.

2. Developing methods to create features for other network protocols such as UDP, ICMP, DNS, IRC and represent them by SoNSTAR. This ensures that a larger amount of representative traffic is covered for the development and testing of network traffic behaviour. The development could also include ICMP and TCP event interaction.

3. Exploiting the very high potential that SoNSTAR has to represent SCADA systems because of their unique nature. If their normal behaviour patterns are tested, zero-day vulnerability mitigation can be increased.

4. Further studies could be performed to evaluate what SoNSTAR could provide to the educational process to enable students to observe traffic in a simple and meaningful way as well as how it could make a contribution towards providing real-time interaction with protocol mechanisms.

5. Examining the use of sonification on router logs and hard disk storage logs for security purposes. Since logs have existed for a long time, it would seem logical to develop sonification tools to help analysts in post-mortem and real-time analysis.

6. Further research is required on botnets and on SoNSTAR integration with IDSs through developing new features to be used in IDSs. A botnet adopts various behaviours and could be recognised as malicious behaviours rather than classified as a botnet.

7. Studying botnet characteristic in the DNS protocol using SoNSTAR.

8. Further research is required into the real-time visualisation of the IP flow reduction technique. Since previous work did not have access to the IP flow concept used in SoNSTAR when for carrying out visualisations so it will be of benefit to consider. Since the main aim of this research was to use sonification to allow real-time monitoring, the visualisation of aggregated traffic flows was not investigated.

References

[1] Adams, M., Cox, T., Moore, G., Croxford, B., Refaee, M., and Sharples, S. 'Sustainable Soundscapes: Noise Policy and the Urban Experience'. *Urban Studies*, **43**(13):pp. 2385–2398, 2006.

[2] Aggarwal, C. C. *Data Classification: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition, 2014. ISBN 1466586745, 9781466586741.

[3] Alshammari, R. and Zincir-Heywood, A. N. 'Can Encrypted Traffic Be Identified Without Port Numbers, IP Addresses and Payload Inspection?' *Computer Networks*, **55**(6):pp. 1326–1350, 2011.

[4] Angerman, W. S. *Coming Full Circle with Boyd's OODA Loop Ideas: An Analysis of Innovation Diffusion and Evolution*. Master's thesis, Airforce Institute of Technology, Wright-Patterson AFB, Ohio, USA, March 2004.

[5] Axelsson, S. and Sands, D. *Understanding Intrusion Detection Through Visualization*. Springer Science & Business Media, 2006.

[6] Aycock, J. *Computer Viruses and Malware*. Springer Science & Business Media, 2006.

[7] Backhaus, S., Bent, R., Bono, J., Lee, R., Tracey, B., Wolpert, D., Xie, D., and Yildiz, Y. 'Cyber-physical security: A game theory model of humans interacting over control systems'. *IEEE Transactions on Smart Grid*, **4**(4):pp. 2320–2327, 2013.

[8] Baier, G., Hermann, T., Sahle, S., and Stephani, U. 'Sonified Epilectic Rhythms'. In T. Stockman, L. V. Nickerson, C. Frauenberger, A. D. N. Edwards, and D. Brock (Eds.), *ICAD 2006 - the 12th Meeting of the International Conference on Auditory Display*, pp. 148–151. London, UK, 20–23 June 2006.

[9] Bakker, S., Van Den Hoven, E., and Eggen, B. 'Exploring interactive systems using peripheral sounds'. In *International Workshop on Haptic and Audio Interaction Design*, pp. 55–64. Springer, 2010.

[10] Ballora, M., Cole, R. J., Kruesi, H., Greene, H., Monahan, G., and Hall, D. L. 'Use of Sonification in the Detection of Anomalous Events'. In *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2012*, volume 8407, p. 84070S. International Society for Optics and Photonics, 2012.

[11] Ballora, M., Giacobe, N. A., and Hall, D. L. 'Songs of Cyberspace: An Update on Sonifications of Network Traffic to Support Situational Awareness'. In *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2011*, volume 8064, p. 80640P. International Society for Optics and Photonics, 2011.

[12] Ballora, M. and Hall, D. L. '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*, volume 7709, p. 77090J. International Society for Optics and Photonics, 2010.

[13] Barber, J. R., Crooks, K. R., and Fristrup, K. M. 'The Costs of Chronic Noise Exposure for Terrestrial Organisms'. *Trends in Ecology & Evolution*, **25**(3):pp. 180–189, 2010.

[14] Barford, P., Kline, J., Plonka, D., and Ron, A. 'A Signal Analysis of Network Traffic Anomalies'. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment*, pp. 71–82. ACM, 2002.

[15] Barford, P. and Yegneswaran, V. 'An Inside Look at Botnets'. In M. Christodorescu, S. Jha, D. Maughan, D. Song, and C. Wang (Eds.), *Malware Detection*, pp. 171–191. Boston, MA: Springer US, 2007.

[16] Bass, T. 'Intrusion Detection Systems and Multisensor Data Fusion'. *Communications of the ACM*, **43**(4):pp. 99–105, 2000.

[17] Becher, M. *Web Application Firewalls*. VDM Verlag, 2007.

[18] Bernaille, L. and Teixeira, R. 'Early Recognition of Encrypted Applications'. In S. Uhlig, K. Papagiannaki, and O. Bonaventure (Eds.), *Passive and Active Network Measurement*, pp. 165–175. Berlin, Heidelberg: Springer, 2007.

[19] Bly, S. 'Sound and computer information presentation'. Technical report, Lawrence Livermore National Lab., CA (USA); California Univ., Davis (USA), 1982.

[20] Bowman, W. B. 'System and Method for Detecting Fraudulent Network Usage Patterns Using Real-Time Network Monitoring'. US Patent 5,627,886/ Google Patents., May 6 1997.

[21] Bradley, N., Alvarez, M., McMillen, D., and Craig, S. 'Reviewing a Year of Serious Data Breaches, Major Attacks and New Vulnerabilities'. Cyber Security Intelligence Index, IBM X-Force Research, 2016.

[22] Brehmer, B. 'The Dynamic OODA Loop: Amalgamating Boyd's OODA Loop and the Cybernetic Approach to Command and Control'. In *Proceedings of the 10th International Command and Control Research Technology Symposium*. 2005.

[23] Brewster, S. A., Wright, P. C., and Edwards, A. D. N. 'A Detailed Investigation into the Effectiveness of Earcons'. In G. Kramer (Ed.), *Auditory Display*, *Santa FE Institute Studies in the Sciences of Complexity-Proceedings*, volume XVIII, pp. 471–498. Reading, MA: Addison-Wesley, 1994.

[24] Brown, C., Cowperthwaite, A., Hijazi, A., and Somayaji, A. 'Analysis of the 1999 DARPA/Lincoln Laboratory IDS Evaluation Data with NetADHICT'. In *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium On*, pp. 1–7. IEEE, 2009.

[25] Brownlee, N., Mills, C., and Ruth, G. 'Traffic Flow Measurement: Architecture'. https://tools.ietf.org/html/rfc2722, 1999.

[26] CAIDA Center for Applied Internet Data Analysis. 'The CAIDA DDoS Attack Dataset'. http://www.caida.org/data/overview/, 2007.

[27] Calyam, P., Krymskiy, D., Sridharan, M., and Schopis, P. 'Active and Passive Measurements on Campus, Regional and National Network Backbone Paths'. In *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference On*, pp. 537–542. IEEE, 2005.

[28] Carrier, B. 'Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers'. *International Journal of Digital Evidence*, **1**(4):pp. 1–12, 2003.

[29] Caruso, R. D. 'Personal Computer Security: Part 1. Firewalls, Antivirus Software, and Internet Security Suites 1'. *RadioGraphics*, **23**(5):pp. 1329–1337, 2003.

[30] Casey, E. *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet*. Academic press, 2011.

[31] Cavelty, M. D. and Mauer, V. *Power and Security in the Information Age: Investigating the Role of the State in Cyberspace*. Routledge, 2016.

[32] Chafe, C. and Leistikow, R. 'Levels of Temporal Resolution in Sonification of Network Performance'. In J. Hiipakka, N. Zacharov, and T. Takala (Eds.), *ICAD 2001 7th International Conference on Auditory Display*, pp. 50–55. Espoo, Finland: ICAD, 29 July–1 August 2001.

[33] Chambers, J., Mathews, M., and Moore, F. 'Auditory data inspection'. *Report TM*, pp. 74–122, 1974.

[34] Cisco. 'IOS Netflow Technology Data Sheet'. http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/iosnf_ds.htm/, 2016.

[35] Clarke-Salt, J. *SQL Injection Attacks and Defense*. Elsevier, 2009.

[36] Cooke, E., Jahanian, F., and McPherson, D. 'The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets.' *SRUTI*, **5**:pp. 6–6, 2005.

[37] Corero Network Security. 'A Network's New First Line of Defense'. http://www.corero.com/resources/files/whitepapers/cns_whitepaper_firstlineofdefense.pdf/, 2013.

[38] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and Cheshire, S. 'Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry'. https://tools.ietf.org/html/rfc6335, 2011.

[39] Criado, R., Flores, J., Hernández-Bermejo, B., Pello, J., and Romance, M. 'Effective Measurement of Network Vulnerability Under Random and Intentional Attacks'. *Journal of Mathematical Modelling and Algorithms*, **4**(3):pp. 307–316, 2005.

[40] d'Albe, E. F. 'On a type-reading optophone'. *Proc. R. Soc. Lond. A*, **90**(619):pp. 373–375, 1914.

[41] Debashi, M. and Vickers, P. 'Nuson-SoNSTAR: Sonification of Networks for SiTuational AwaReness'. https://github.com/nuson/SoNSTAR. DOI: 10.5281/zenodo.1072535, 2017.

[42] Degara, N., Hunt, A., and Hermann, T. 'Interactive Sonification [Guest editors' introduction]'. *IEEE MultiMedia*, **22**(1):pp. 20–23, Jan.-Mar. 2015. ISSN 1070-986X.

[43] Dejmal, S., Fern, A., and Nguyen, T. P. 'Reinforcement Learning for Vulnerability Assessment in Peer-to-Peer Networks.' In *AAAI*, pp. 1655–1662. 2008.

[44] Di Pietro, R. and Mancini, L. V. *Intrusion Detection Systems*. Springer, 2008.

[45] Endsley, M. R. 'Toward a Theory of Situation Awareness in Dynamic Systems'. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, **37**(1):pp. 32–64, 1995.

[46] Eslahi, M., Salleh, R., and Anuar, N. B. 'Bots and Botnets: An Overview of Characteristics, Detection and Challenges'. In *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference On*, pp. 349–354. IEEE, 2012.

[47] Fairfax, T., Laing, C., and 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), pp. 334–349. IGI Global, July 2014.

[48] Fall, K. R. and Stevens, W. R. *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley, 2011.

[49] Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Khan, M., Moll, D., Rockell, R., Seely, T., and Diot, S. 'Packet-Level Traffic Measurements from the Sprint IP Backbone'. *IEEE Network*, **17**(6):pp. 6–16, 2003.

[50] Frysinger, S. P. 'A brief history of auditory data representation to the 1980s'. Georgia Institute of Technology, 2005.

[51] Fung, C. 'Collaborative Intrusion Detection Networks and Insider Attacks'. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, **2**(1):pp. 63–74, 2011.

[52] Gadge, J. and Patil, A. A. 'Port Scan Detection'. In *Networks, 2008. ICON 2008. 16th IEEE International Conference On*, pp. 1–6. IEEE, 2008.

[53] Garcia-Ruiz, M. A., Block, A. E., Martin, M. V., and ElSeoud, S. 'Auditory Display As a Tool for Teaching Network Intrusion Detection.' *iJET*, **3**(2):pp. 59–62, 2008.

[54] García-Ruiz, M. Á., Kapralos, B., and Vargas Martin, M. 'Towards Multimodal Interfaces for Intrusion Detection'. In *Audio Engineering Society Convention 122*. Audio Engineering Society, 2007.

[55] Gaver, W. W. 'Auditory Icons: Using Sound in Computer Interfaces'. *Human-Computer Interaction*, **2**(2):pp. 167–177, 1986.

[56] Gaver, W. W., Smith, R. B., and O'Shea, T. 'Effective sounds in complex systems: The ARKola simulation'. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, pp. 85–90. ACM, 1991.

[57] Geerthidevi, K. G., Prakash, T. S., and Tharani, S. 'Social Network Based Security Schema for Botnet Detection and Prevention'. *International Journal Of Engineering And Computer Science*, **4**(6), 2015.

[58] Ghorbani, A. A., Lu, W., and Tavallaee, M. 'Network Attacks'. In *Network Intrusion Detection and Prevention*, pp. 1–25. Springer, 2010.

[59] Gilfix, M. and Couch, A. L. 'Peep (the Network Auralizer): Monitoring Your Network with Sound.' In *14th System Administration Conference (LISA 2000)*, pp. 109–117. New Orleans, Louisiana, USA: The USENIX Association, 3–8 December 2000.

[60] Giot, R. and Courbe, Y. 'InteNtion–Interactive Network Sonification'. In M. A. Nees, B. N. Walker, and J. Freeman (Eds.), *Proceedings of the 18th International Conference on Auditory Display (ICAD 2012)*, pp. 235–236. Georgia Institute of Technology, 2012.

[61] Goldman, J. and Maret, S. *Intelligence and Information Policy for National Security: Key Terms and Concepts*. Rowman & Littlefield, 2016.

[62] Goodall, J. R. 'Introduction to Visualization for Computer Security'. In J. R. Goodall, G. Conti, and K.-L. Ma (Eds.), *VizSEC 2007: Proceedings of the Workshop on Visualization for Computer Security*, pp. 1–17. Springer, 2008.

[63] Gopinath, M. C. *Auralization of Intrusion Detection System Using JListen*. Master's thesis, Birla Institute of Technology and Science, Pilani (Rajasthan), India, May 2004.

[64] Grond, F. and Hermann, T. 'Interactive Sonification for Data Exploration: How Listening Modes and Display Purposes Define Design Guidelines'. *Organised Sound*, **19**(1):pp. 41–51, 2014.

[65] Gupta, B. B., Arachchilage, N. A. G., and Psannis, K. E. 'Defending Against Phishing Attacks: Taxonomy of Methods, Current Issues and Future Directions'. *Telecommunication Systems*, **67**(2):pp. 247–267, February 2018.

[66] Gupta, B. B., Tewari, A., Jain, A. K., and Agrawal, D. P. 'Fighting Against Phishing Attacks: State of the Art and Future Challenges'. *Neural Computing and Applications*, **28**(12):pp. 3629–3654, December 2017.

[67] Hart, S. G. and Staveland, L. E. 'Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research'. *Advances in psychology*, **52**:pp. 139–183, 1988.

[68] Hermann, T., Drees, J. M., and Ritter, H. 'Broadcasting auditory weather reports-a pilot project'. Georgia Institute of Technology, 2003.

[69] Hermann, T. and Hunt, A. 'Guest Editors' Introduction: An Introduction to Interactive Sonification'. *IEEE Multimedia*, **12**(2):pp. 20–24, 2005.

[70] Hermann, T., Hunt, A., and Neuhoff, J. G. *The Sonification Handbook*. Logos Verlag Berlin, 2011.

[71] Hermann, T. and Ritter, H. 'Listen to Your Data: Model-Based Sonification for Data Analysis'. In G. E. Lasker (Ed.), *Advances in Intelligent Computing and Multimedia Systems*, pp. 189–194. Baden-Baden, Germany: Int. Inst. for Advanced Studies in System research and cybernetics, August 1999.

[72] Hildebrandt, T. 'Towards Enhancing Business Process Monitoring with Sonification'. In *Business Process Management Workshops*, pp. 529–536. Springer, 2014.

[73] Hildebrandt, T. and Rinderle-Ma, S. 'Toward a sonification concept for business process monitoring'. 2013.

[74] Hildebrandt, T. and Rinderle-Ma, S. 'Server Sounds and Network Noises'. In *Cognitive Infocommunications (CogInfoCom), 2015 6th IEEE International Conference On*, pp. 45–50. IEEE, 2015.

[75] Hunt, A. and Hermann, T. 'Guest Editors' Introduction: An Introduction to Interactive Sonification'. *IEEE MultiMedia*, **12**:pp. 20–24, 04 2005. ISSN 1070-986X.

[76] Hunt, A. and Hermann, T. 'Interactive Sonification'. In T. Hermann, A. D. Hunt, and J. Neuhoff (Eds.), *The Sonification Handbook*, pp. 273–298. Berlin: Logos Verlag, 2011. ISBN 978-3-8325-2819-5.

[77] Hussain, A., Heidemann, J., and Papadopoulos, C. 'A Framework for Classifying Denial of Service Attacks'. In *Proceedings of the 2003 Conference on Applications,*

Technologies, Architectures, and Protocols for Computer Communications, pp. 99–110. ACM, 2003.

[78] Hussein, S. M., Ali, F. H. M., and Kasiran, Z. 'Evaluation effectiveness of hybrid IDs using snort with naive Bayes to detect attacks'. In *Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on*, pp. 256–260. IEEE, 2012.

[79] Hutchins, E. M., Cloppert, M. J., and Amin, R. M. 'Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains'. In *Proceedings of the 6th International Conference on Information Warfare and Security*, pp. 113–125. George Washington University, Washington DC, USA, 17–18 March 2011.

[80] InMon Corporation. 'SFlow Tool'. http://www.sflow.org/, 2017.

[81] Jain, M. and Dovrolis, C. 'Pathload: A Measurement Tool for End-To-End Available Bandwidth'. In *Proceedings of Passive and Active Measurements (PAM) Workshop*, pp. 14–25. 2002.

[82] Jajodia, S., Liu, P., Swarup, V., and Wang, C. (Eds.). *Cyber Situational Awareness*. Springer, 2010.

[83] Jung, J. *Real-Time Detection of Malicious Network Activity Using Stochastic Models*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, June 2006.

[84] Jung, J., Krishnamurthy, B., and Rabinovich, M. 'Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites'. In *Proceedings of the 11th International Conference on World Wide Web*, pp. 293–304. ACM, 2002.

[85] Jurdak, R., Ruzzelli, A. G., Barbirato, A., and Boivineau, S. 'Octopus: Monitoring, Visualization, and Control of Sensor Networks'. *Wireless Communications and Mobile computing*, **11**(8):pp. 1073–1091, 2011.

[86] Kang, J. and Zhang, J.-Y. 'Application Entropy Theory to Detect New Peer-To-Peer Botnet with Multi-Chart CUSUM'. In *Electronic Commerce and Security, 2009. ISECS'09. Second International Symposium On*, volume 1, pp. 470–474. IEEE, 2009.

[87] Karim, A., Salleh, R. B., Shiraz, M., Shah, S. A. A., Awan, I., and Anuar, N. B. 'Botnet Detection Techniques: Review, Future Trends, and Issues'. *Journal of Zhejiang University SCIENCE C*, **15**(11):pp. 943–983, 2014.

[88] Karlaftis, M. G. and Vlahogianni, E. I. 'Statistical methods versus neural networks in transportation research: Differences, similarities and some insights'. *Transportation Research Part C: Emerging Technologies*, **19**(3):pp. 387–399, 2011.

[89] Katz, B. F. G. and Marentakis, G. 'Advances in auditory display research'. *Journal on Multimodal User Interfaces*, **10**(3):pp. 191–193, Sep 2016. ISSN 1783-8738.

[90] Kim, I., Choi, H., and Lee, H. 'Botnet Visualization Using DNS Traffic'. In *Proc. of WISA*. 2008.

[91] Kimoto, M. and Ohno, H. 'Design and Implementation of Stetho Network Sonification System'. In *Proceedings of the 2002 International Computer Music Conference*, pp. 273–279. 2002.

[92] Kirubavathi, G. and Anitha, R. 'Botnet Detection Via Mining of Traffic Flow Characteristics'. *Computers & Electrical Engineering*, **50**:pp. 91–101, 2016.

[93] Komlodi, A., Goodall, J. R., and Lutters, W. G. 'An Information Visualization Framework for Intrusion Detection'. In *CHI'04 Extended Abstracts on Human Factors in Computing Systems*, p. 1743. ACM, 2004.

[94] Kopetz, H. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Springer Science & Business Media, 2011.

[95] Kozierok, C. M. *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*. No Starch Press, 2005.

[96] Kramer, G. (Ed.). *Auditory Display: Sonification, Audification, and Auditory Interfaces*, *Santa Fe Institute, Studies in the Sciences of Complexity Proceedings*, volume XVIII. Reading, MA: Addison-Wesley, 1994.

[97] Kramer, G. 'Preface'. In G. Kramer (Ed.), *Auditory Display*, *Santa Fe Institute, Studies in the Sciences of Complexity Proceedings*, volume XVIII, pp. xxiii–xxxviii. Reading, MA: Addison-Wesley, 1994.

[98] Kramer, G. 'Some Organizing Principles for Representing Data with Sound'. In G. Kramer (Ed.), *Auditory Display*, *Santa Fe Institute, Studies in the Sciences of Complexity Proceedings*, volume XVIII, pp. 185–222. Reading, MA: Addison-Wesley, 1994.

[99] Kramer, G., Walker, B., Bonebright, T., Cook, P., Flowers, J. H., Miner, N., and Neuhoff, J. 'Sonification Report: Status of the Field and Research Agenda'. *Faculty Publications, Department of Psychology. 444*, 2010.

[100] Kramer, G. and Walker, B. N. 'Sound Science: Marking Ten International Conferences on Auditory Display'. *ACM Trans. Appl. Percept.*, **2**(4):pp. 383–388, October 2005. ISSN 1544-3558.

[101] Laing, C. and Vickers, P. 'Context Informed Intelligent Information Infrastructures for Better Situational Awareness'. In *Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), 2015 International Conference*, pp. 1–7. IEEE, 2015.

[102] Lakkaraju, K., Yurcik, W., and Lee, A. J. '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*, pp. 65–72. ACM, 2004.

[103] Lee, W., Stolfo, S. J., and Mok, K. W. 'A Data Mining Framework for Building Intrusion Detection Models'. In *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium On*, pp. 120–132. IEEE, 1999.

[104] Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., and Das, K. 'The 1999 DARPA Off-Line Intrusion Detection Evaluation'. *Computer networks*, **34**(4):pp. 579–595, 2000.

[105] Liu, L. and Özsu, M. T. (Eds.). *Encyclopedia of Database Systems*. Springer Berlin, Heidelberg, Germany, 2009.

[106] Mahajan, S. 'Reinforcement Learning: A Review from a Machine Learning Perspective'. *International Journal*, **4**(8), 2014.

[107] Malandrino, D., Mea, D., Negro, A., Palmieri, G., and Scarano, V. 'NeMoS: Network Monitoring with Sound'. In E. Brazil and B. Shinn-Cunningham (Eds.), *Proceedings of the 2003 International Conference on Auditory Display, Boston, MA, USA*, pp. 251–254. Georgia Institute of Technology, Boston, MA: ICAD, 2003.

[108] Mancuso, V. F., Greenlee, E. T., Funke, G., Dukes, A., Menke, L., Brown, R., and Miller, B. 'Augmenting Cyber Defender Performance and Workload Through Sonified Displays'. *Procedia Manufacturing*, **3**:pp. 5214–5221, 2015.

[109] Markou, M. and Singh, S. 'Novelty detection: a review?part 1: statistical approaches'. *Signal processing*, **83**(12):pp. 2481–2497, 2003.

[110] Miller, S. and Busby-Earle, C. 'The role of machine learning in botnet detection'. In *Internet Technology and Secured Transactions (ICITST), 2016 11th International Conference for*, pp. 359–364. IEEE, 2016.

[111] Morgan, C. 'SharpPcap: Fully Managed, Cross Platform (Windows, Mac, Linux) .NET Library for Capturing Packets'. https://github.com/chmorgan/sharppcap, 2017.

[112] Mukkamala, S., Sung, A., and Abraham, A. 'Cyber Security Challenges: Designing Efficient Intrusion Detection Systems and Antivirus Tools'. *Vemuri, V. Rao, Enhancing Computer Security with Smart Technology.(Auerbach, 2006)*, pp. 125–163, 2005.

[113] Neuhoff, J. G., Wayand, J., and Kramer, G. 'Pitch and Loudness Interact in Auditory Displays: Can the Data Get Lost in the Map?' *Journal of Experimental Psychology: Applied*, **8**(1):p. 17, 2002.

[114] Northcutt, S., Novak, J., and McLachlan, D. *Network Intrusion Detection: An Analyst's Handbook*. New Riders Publishing, 2nd edition, 2000.

[115] Ohsita, Y., Shingo, A., and Murata, M. 'Detecting Distributed Denial-Of-Service Attacks by Analyzing TCP SYN Packets Statistically'. *IEICE Transactions on Communications*, **89**(10):pp. 2868–2877, 2006.

[116] Olson, D. L. and Delen, D. *Advanced Data Mining Techniques*. Springer Science & Business Media, 2008.

[117] Onwubiko, C. 'Functional Requirements of Situational Awareness in Computer Network Security'. In *Intelligence and Security Informatics, 2009. ISI'09. IEEE International Conference On*, pp. 209–213. IEEE, 2009.

[118] Onwubiko, C. and Owens, T. (Eds.). *Situational Awareness in Computer Network Defense: Principles, Methods and Applications*. IGI Global, 2012.

[119] Panchen, S., Phaal, P., and McKee, N. 'InMon Corporation's SFlow: A Method for Monitoring Traffic in Switched and Routed Networks'. https://tools.ietf.org/html/rfc3176, 2001.

[120] Pauletto, S. and Hunt, A. 'Interactive Sonification of Complex Data'. *International Journal of Human-Computer Studies*, **67**(11):pp. 923–933, 2009.

[121] Pijanowski, B. C., Villanueva-Rivera, L. J., Dumyahn, S. L., Farina, A., Krause, B. L., Napoletano, B. M., Gage, S. H., and Pieretti, N. 'Soundscape Ecology: The Science of Sound in the Landscape'. *BioScience*, **61**(3):pp. 203–216, 2011.

[122] Polikar, R., Upda, L., Upda, S. S., and Honavar, V. 'Learn++: An incremental learning algorithm for supervised neural networks'. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, **31**(4):pp. 497–508, 2001.

[123] Pollack, I. and Ficks, L. 'Information of elementary multidimensional auditory displays'. *The Journal of the Acoustical Society of America*, **26**(2):pp. 155–158, 1954.

[124] Powers, D. M. W. 'Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation'. *Journal of Machine Learning Technologies*, **2**(1):pp. 37–63, 2011.

[125] Proctor, P. E. *Practical Intrusion Detection Handbook*. Prentice Hall PTR, 2000.

[126] Ramzan, Z. 'Phishing Attacks and Countermeasures'. *Handbook of Information and Communication Security*, pp. 433–448, 2010.

[127] Ranjan, S., Robinson, J., and Chen, F. 'Machine learning based botnet detection using real-time connectivity graph based traffic features', 2014.

[128] Rauterberg, M. and Styger, E. 'Positive effects of sound feedback during the operation of a plant simulator'. In *International Conference on Human-Computer Interaction*, pp. 35–44. Springer, 1994.

[129] RTI International. 'PREDICT: Protected Repository for the Defense of Infrastructure Against Cyber Threats'. http://www.predict.org, 2011.

[130] Rutz, H. H., Vogt, K., and Höldrich, R. 'The SysSon Platform: A Computer Music Perspective of Sonification'. In K. Vogt, A. Andreopoulou, and V. Goudarzi (Eds.), *ICAD 15: Proceedings of the 21st International Conference on Auditory Display*, pp. 188–196. Graz, Austria: Institute of Electronic Music and Acoustics (IEM), University of Music and Performing Arts Graz (KUG), 2015.

[131] Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., Felix, J., and Hakimian, P. 'Detecting P2P Botnets Through Network Behavior Analysis and Machine Learning'. In *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference On*, pp. 174–180. IEEE, 2011.

[132] Sagiroglu, S. and Canbek, G. 'Keyloggers'. *IEEE Technology and Society Magazine*, **28**(3), 2009.

[133] Schafer, R. M. *The Tuning of the World*. Random House, 1977.

[134] Schedel, M. and Worrall, D. R. 'Editorial'. *Organised Sound*, **19**(1):pp. 1–3, 2014.

[135] Schmandt, C. and Vallejo, G. "'Listenin"to domestic enviroments from remote locations'. Georgia Institute of Technology, 2003.

[136] Schuba, C. L., Krsul, I. V., Kuhn, M. G., Spafford, E. H., Sundaram, A., and Zamboni, D. 'Analysis of a Denial of Service Attack on TCP'. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pp. 208–223. IEEE, 1997.

[137] Seo, I., Lee, H., and Han, S. C. 'Cylindrical Coordinates Security Visualization for Multiple Domain Command and Control Botnet Detection'. *Computers & Security*, **46**:pp. 141–153, 2014.

[138] Shah, K., Bohacek, S., and Broido, A. 'Feasibility of Detecting TCP-SYN Scanning at a Backbone Router'. In *American Control Conference, 2004. Proceedings of the 2004*, volume 2, pp. 988–995. IEEE, 2004.

[139] Shahrestani, A., Feily, M., Ahmad, R., and Ramadass, S. 'Architecture for Applying Data Mining and Visualization on Network Flow for Botnet Traffic Detection'. In *Computer Technology and Development, 2009. ICCTD'09. International Conference On*, volume 1, pp. 33–37. IEEE, 2009.

[140] Sikorski, M. and Honig, A. *Practical Malware Analysis: The Hands-on Guide to Dissecting Malicious Software*. No Starch Press, 2012.

[141] Sommer, R. and Paxson, V. 'Enhancing byte-level network intrusion detection signatures with context'. In *Proceedings of the 10th ACM conference on Computer and communications security*, pp. 262–271. ACM, 2003.

[142] Soniya, B. and Wiscy, M. 'Detection of TCP SYN Scanning Using Packet Counts and Neural Network'. In *Signal Image Technology and Internet Based Systems, 2008. SITIS'08. IEEE International Conference On*, pp. 646–649. IEEE, 2008.

[143] Speeth, S. D. 'Seismometer sounds'. *The Journal of the Acoustical Society of America*, **33**(7):pp. 909–916, 1961.

[144] Srinivasan, T., Vijaykumar, V., and 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*, pp. 1–6. IEEE, 2006.

[145] Stallings, W. and Brown, L. *Computer Security: Principles and Practice*. Pearson Education Limited, 2nd edition, 2008.

[146] Stalmans, E. and Irwin, B. 'A Framework for DNS Based Detection and Mitigation of Malware Infections on a Network'. In *Information Security South Africa (ISSA), 2011*, pp. 1–8. IEEE, 2011.

[147] Subramanyam, K., Frank, C. E., and Galli, D. H. 'Keyloggers: The Overlooked Threat to Computer Security'. In *1st Midstates Conference for Undergraduate Research in Computer Science and Mathematics*. 2003.

[148] Sutherland, L. 'Know Your Enemy: Understanding the Motivation Behind Cyberattacks', Security Intelligence. IBM. https://securityintelligence.com, March 2016.

[149] The Shmoo Group. 'DEFCON Dataset'. http://cctf.shmoo.com, 2011.

[150] The Snort Community. 'Snort community rules'. https://snort.org/, 2016.

[151] Thomas, T. M. and Stoddard, D. *Network Security First-Step*. Cisco Press, 2nd edition, 2011.

[152] Tran, Q. T. and Mynatt, E. D. 'Music monitor: Ambient musical data for the home'. *Extended Proceedings of the HOIT*, pp. 85–92, 2000.

[153] Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., and Lin, W.-Y. 'Intrusion detection by machine learning: A review'. *Expert Systems with Applications*, **36**(10):pp. 11994–12000, 2009.

[154] University of New Brunswick. 'ISCX Datasets - the Canadian Institute for Cybersecurity'. http://www.unb.ca/cic/research/datasets/, 2012.

[155] University of Victoria. 'ISOT Botnet Dataset'. http://www.uvic.ca/engineering/ece/isot/datasets/, 2010.

[156] Vickers, P. 'Sonification for Process Monitoring'. In T. Hermann, A. D. Hunt, and J. Neuhoff (Eds.), *The Sonification Handbook*, pp. 455–492. Berlin: Logos Verlag, 2011. ISBN 978-3-8325-2819-5.

[157] Vickers, P. and Alty, J. L. 'CAITLIN: a musical problem auralisation tool to assist novice programmers with debugging'. Georgia Institute of Technology, 1996.

[158] Vickers, P., Laing, C., Debashi, M., and Fairfax, T. 'Sonification Aesthetics and Listening for Network Situational Awareness'. In *SoniHED — Conference on Sonification of Health and Environmental Data*. University of York, 12 September 2014.

[159] Vickers, P., Laing, C., and Fairfax, T. 'Sonification of a Network's Self-Organized Criticality for Real-Time Situational Awareness'. *Displays*, **47**:pp. 12–24, April 2017.

[160] Vickers, P., Worrall, D., and So, R. 'Preface to the Special Issue on Sonification'. *Displays*, **47**:p. 1, April 2017.

[161] Walker, B. N. and Kramer, G. 'Ecological Psychoacoustics and Auditory Displays: Hearing, Grouping, and Meaning Making'. In J. G. Neuhoff (Ed.), *Ecological Psychoacoustics*, pp. 150–175. Elsevier Academic Press, 2004.

[162] Warkentin, M. and Willison, R. 'Behavioral and policy issues in information systems security: the insider threat'. *European Journal of Information Systems*, **18**(2):pp. 101–105, 2009.

[163] Williamson, C. 'Internet Traffic Measurement'. *IEEE Internet Computing*, **5**(6):pp. 70–74, 2001.

[164] Wireshark Foundation. 'Wireshark Tool'. https://www.wireshark.org/, 2017.

[165] Wolf, K. E. and Fiebrink, R. 'SonNet: A Code Interface for Sonifying Computer Network Data'. In *NIME, 13 — 13th International Conference on New Interfaces for Musical Expression*, pp. 503–506. 2013.

[166] Wolf, K. E., Gliner, G., and 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, pp. 97–100. Atlanta, GA: ACM, 2015.

[167] Worrall, D. 'Realtime Sonification and Visualisation of Network Metadata'. In K. Vogt, A. Andreopoulou, and V. Goudarzi (Eds.), *Proceedings of the 21st International Conference on Auditory Display (ICAD 2015)*, pp. 337–339. Graz, Austria: Institute of Electronic Music and Acoustics (IEM), University of Music and Performing Arts Graz (KUG), 2015.

[168] Yang, J. and Hunt, A. 'Real-Time Sonification of Biceps Curl Exercise Using Muscular Activity and Kinematics'. In K. Vogt, A. Andreopoulou, and V. Goudarzi (Eds.), *Proceedings of the 21st International Conference on Auditory Display (ICAD 2015)*, pp. 289–293. Graz, Austria: Institute of Electronic Music and Acoustics (IEM), University of Music and Performing Arts Graz (KUG), 2015.

[169] Yegneswaran, V., Barford, P., and Ullrich, J. 'Internet Intrusions: Global Characteristics and Prevalence'. *ACM SIGMETRICS Performance Evaluation Review*, **31**(1):pp. 138–147, 2003.

[170] Yin, X., Yurcik, W., Treaster, M., Li, Y., and Lakkaraju, K. 'VisFlowConnect: Netflow Visualizations of Link Relationships for Security Situational Awareness'. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, pp. 26–34. ACM, 2004.

[171] Zhang, J., Perdisci, R., Lee, W., Sarfraz, U., and Luo, X. 'Detecting Stealthy P2P Botnets Using Statistical Traffic Fingerprints'. In *Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference On*, pp. 121–132. IEEE, 2011.

[172] Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., and Garant, D. 'Botnet Detection Based on Traffic Behavior Analysis and Flow Intervals'. *Computers & Security*, **39**:pp. 2–16, 2013.

Appendix A

Consent form 1

The participant consent form for the study described in Chapter 4 is reproduced overleaf.

Faculty of Engineering and Environment

RESEARCH PARTICIPANT CONSENT FORM

Name of participant	
Researcher's name	
Programme of study	
Supervisor's name	

Brief description of nature of research and involvement of participant: The research is about increasing the situational awareness of real time network monitoring tools and because situational awareness of network activity needs to be maintained to ensure an appropriate response to attacks and the efficient management of network resources. We researchers want to learn from your experience of using sonification (SoNSTAR) tools for network monitoring purpose to support situational awareness. AS participant, you will be asked to perform one tasks of network monitoring using both tools and will be asked to fill out questionnaire designed to collect specific information for each task condition.

****Statement of participant consent (please tick as appropriate)**

I confirm that:

I have been briefed about this research project and its purpose and agree to participate ☐

I have been given the opportunity to ask questions about the project and my participation. ☐

I voluntarily agree to participate in the project. ☐

I understand I can withdraw at any time without giving reasons and that I will not be penalised for withdrawing nor will I be questioned on why I have withdrawn ☐

I have discussed any requirement for anonymity or confidentiality with the researcher ☐

I agree to being audio recorded/filmed/photographed ☐

****Specific requirements for anonymity, confidentiality, data storage, retention and destruction**

Participant:
Signed: …… Date: ……

Researcher:
Signed: …… Date: ……

Appendix B

Consent form 2

The participant consent form for the experiment discussed in Chapter 5 is reproduced overleaf.

Faculty of Engineering and Environment

RESEARCH PARTICIPANT CONSENT FORM

Name of participant	
Researcher's name	
Programme of study	
Supervisor's name	

Brief description of nature of research and involvement of participant: The research is about increasing the situational awareness of real time network monitoring tools and because situational awareness of network activity needs to be maintained to ensure an appropriate response to attacks and the efficient management of network resources. We researchers want to learn from your experience of using visualisation only (Snort) and sonification (SoNSTAR) tools for network monitoring purpose to support situational awareness. AS participant, you will be asked to perform three tasks of network monitoring using both tools and will be asked to fill out questionnaire designed to collect specific information for each task condition.

****Statement of participant consent (please tick as appropriate)**

I confirm that:

I have been briefed about this research project and its purpose and agree to participate ☐

I have been given the opportunity to ask questions about the project and my participation. ☐

I voluntarily agree to participate in the project. ☐

I understand I can withdraw at any time without giving reasons and that I will not be penalised for withdrawing nor will I be questioned on why I have withdrawn ☐

I have discussed any requirement for anonymity or confidentiality with the researcher ☐

I agree to being audio recorded/filmed/photographed ☐

****Specific requirements for anonymity, confidentiality, data storage, retention and destruction**

Participant:
Signed: …… Date: ……

Researcher:
Signed: …… Date: ……

Appendix C

SoNSTAR questionnaire

The participant questionnaire for the study described in Chapter 4 is reproduced on the following two pages.

Training and Guidelines Sheet (SoNSTAR)

This training and guideline sheet helps to learn how the system (SoNSTAR) works . It facilitates the learning of the sounds used in this system and to facilitate the distinction between them.
Please note that the experiment contain one task condition and you should fill the Monitoring and Detection section while performing the task.

About you

1. **Your Gender**: ○ Male ○ Female
2. **What is your level of education?** ○ BSc ○ MSc ○ PhD ○ Others _____
3. **What is your specialty?** ○ Computer Science ○ IT ○ Others _____
4. **What is your studying department?** ○ Computer Science and Digital Technologies ○ Others _____
5. **What is your year of study?** ○ first ○ second ○ third ○ forth

Monitoring and Detection Task:

6. **You will be expecting sounds behaviour changes every one minutes through the next 7 minutes. Please check boxes according sounds heard form State1 to State 7**.

No.	Sound	State 1	State 2	State 3	State 4	State 5	State 6	State 7
1	Birds	☐	☐	☐	☐	☐	☐	☐
2	Rain	☐	☐	☐	☐	☐	☐	☐
3	Rain on Roof	☐	☐	☐	☐	☐	☐	☐
4	Heavy Rain	☐	☐	☐	☐	☐	☐	☐
5	Rain and Thunder	☐	☐	☐	☐	☐	☐	☐
6	Fountain	☐	☐	☐	☐	☐	☐	☐
7	Beach	☐	☐	☐	☐	☐	☐	☐
8	Creek	☐	☐	☐	☐	☐	☐	☐
9	Grass Hopper	☐	☐	☐	☐	☐	☐	☐
10	Spring Peeper	☐	☐	☐	☐	☐	☐	☐
11	Crickets	☐	☐	☐	☐	☐	☐	☐
12	Owls	☐	☐	☐	☐	☐	☐	☐
13	Sheep	☐	☐	☐	☐	☐	☐	☐
14	Frogs	☐	☐	☐	☐	☐	☐	☐
15	Horse Snort	☐	☐	☐	☐	☐	☐	☐
16	Wolves	☐	☐	☐	☐	☐	☐	☐
17	Wind on Grass	☐	☐	☐	☐	☐	☐	☐
18	Wind	☐	☐	☐	☐	☐	☐	☐
19	Snow Storm	☐	☐	☐	☐	☐	☐	☐
20	Walk in Snow	☐	☐	☐	☐	☐	☐	☐
21	Fire	☐	☐	☐	☐	☐	☐	☐

Monitoring Evaluation Task:

Please evaluate SoNSTAR (Sonification)

7a. Mental Demand Rate 1 ○—○—○—○—○—○—○—○—○—○ 10

7b. Temporal Demand Rate 1 ○—○—○—○—○—○—○—○—○—○ 10

7c. Physical Demand Rate 1 ○—○—○—○—○—○—○—○—○—○ 10

7d. Performance Rate 1 ○—○—○—○—○—○—○—○—○—○ 10

7e. Effort Rate 1 ○—○—○—○—○—○—○—○—○—○ 10

7f. Frustration Rate 1 ○—○—○—○—○—○—○—○—○—○ 10

7g. Detection Confidence Rate 1 ○—○—○—○—○—○—○—○—○—○ 10

7h. Ease of Use Rate 1 ○—○—○—○—○—○—○—○—○—○ 10

7i. Sound Fatigue Rate 1 ○—○—○—○—○—○—○—○—○—○ 10

Please evaluate the sonification

7b. Aesthetics horrible ○—○—○—○—○—○—○—○—○—○ fantastic

7b. Annoyance horrible ○—○—○—○—○—○—○—○—○—○ fantastic

About this Experiment

8a. Please describe your first impression.

8b. In case you would like some more lines to write, here they are:

Thank you for your feedback and participation

Appendix D

SoNSTAR vs Snort questionnaire

The participant questionnaire for the SoNSTAR vs Snort study described in Chapter 5 is reproduced on the following two pages.

Sonification vs. Visualisation Questionnaire

Welcome to this very important survey with which we researchers want to learn from your experience of using visualisation only (Snort) and sonification (SoNSTAR) tools for network monitoring purpose to support situational awareness. Thank you for filling it all out.
Please note that the experiment contain three sections one for each task condition and you should fill the right section for each task.

About you

1. **Your name**: _____
2. **Your Gender**: ○ Male ○ Female
3. **How old are you?** I am _____ years old.
4. **What is your level of education?** I am _____ student.
5. **What is your specialty?** I am _____
6. **What is your studying department?** I am in the _____ .
7. **What is your year of study?** ○ first ○ second ○ third ○ forth
8. **Are you in a good mood right now to take this experiment?** ○ absolutely ○ not really

Monitoring and Detection Tasks:

9. **Please for each task condition check the boxes of Connection testing and attacks**.

Sl.	Task Condition	A- Snort	B- SoNSTAR	C- Snort and SoNSTAR
1.1	ICMP ping	Yes ☐ No ☐	Yes ☐ No ☐	Yes ☐ No ☐
1.2	SYN Related port scan	Yes ☐ No ☐	Yes ☐ No ☐	Yes ☐ No ☐
1.3	FIN port scan	Yes ☐ No ☐	Yes ☐ No ☐	Yes ☐ No ☐
1.4	XMAS port scan	Yes ☐ No ☐	Yes ☐ No ☐	Yes ☐ No ☐
1.5	NULL port scan	Yes ☐ No ☐	Yes ☐ No ☐	Yes ☐ No ☐
1.6	SYN flood	Yes ☐ No ☐	Yes ☐ No ☐	Yes ☐ No ☐
1.7	DDoS or DoS spoofed IPs	Yes ☐ No ☐	Yes ☐ No ☐	Yes ☐ No ☐

Monitoring Evaluation Tasks:

Please evaluate Snort (Visual Only)

1a. Mental Demand Rate 1 □—□—□—□—□—□—□—□—□—□ 10

1b. Temporal Demand Rate 1 □—□—□—□—□—□—□—□—□—□ 10

1c. Physical Demand Rate 1 □—□—□—□—□—□—□—□—□—□ 10

1d. Performance Rate 1 □—□—□—□—□—□—□—□—□—□ 10

1e. Effort Rate 1 □—□—□—□—□—□—□—□—□—□ 10

1f. Frustration Rate 1 □—□—□—□—□—□—□—□—□—□ 10

1g. Detection Confidence Rate 1 □—□—□—□—□—□—□—□—□—□ 10

1h. Ease of Use Rate 1 □—□—□—□—□—□—□—□—□—□ 10

1i. Visual Fatigue Rate 1 □—□—□—□—□—□—□—□—□—□ 10

Please evaluate SoNSTAR (Sonification)

2a. Mental Demand Rate 1 □—□—□—□—□—□—□—□—□—□ 10

2b. Temporal Demand Rate 1 □—□—□—□—□—□—□—□—□—□ 10

2c. Physical Demand Rate 1 □—□—□—□—□—□—□—□—□—□ 10

2d. Performance Rate 1 □—□—□—□—□—□—□—□—□—□ 10

2e. Effort Rate 1 □—□—□—□—□—□—□—□—□—□ 10

2f. Frustration Rate 1 □—□—□—□—□—□—□—□—□—□ 10

2g. Detection Confidence Rate 1 □—□—□—□—□—□—□—□—□—□ 10

2h. Ease of Use Rate 1 □—□—□—□—□—□—□—□—□—□ 10

2i. Sound Fatigue Rate 1 □—□—□—□—□—□—□—□—□—□ 10

Please evaluate best task condition

3. What is best for you to use for detection? □ Snort □ SoNSTAR □ Both Together

Please evaluate the following tools

4a. Snort horrible □—□—□—□—□ fantastic

4b. SoNSTAR horrible □—□—□—□—□ fantastic

About this Experiment

5. Do you like this experiment? □ Yes □ No

6. Is it really worth your future participation? □ Guess so. □ Probably not. □ Don't know.

7a. Please describe your first impression.

_____

_____

7b. In case you would like some more lines to write, here they are:

_____

_____

_____

_____

Thank you for your feedback and participation

Appendix E

Training and guidance (sonification)

The training and guidance document for the study described in Chapter 4 are reproduced on the following two pages.

Training and Guidelines Sheet (SoNSTAR)

This training and guideline sheet helps to learn how the system (SoNSTAR) works . It facilitates the learning of the sounds used in this system and to facilitate the distinction between them.
Please note that the experiment has one monitoring task.

Monitoring Training Task:

1. **Table 1 contains the sounds that are expected to be heard during the task. In this training, we will play sounds one by one. If you need us to repeat the sound, please ask. If you are satisfied with the audio learning please check the box in front of the sound in the Training 1 column.**
2. **We will demonstrate in real time one of the seven attacks used in the experiment, Please check the boxes provided for each sound heard in the Training 2 column.**

Table 1: List of expected sounds

No.	Sound	Training 1	Training 2
1	Birds	☐	☐
2	Rain	☐	☐
3	Rain on Roof	☐	☐
4	Heavy Rain	☐	☐
5	Rain and Thunder	☐	☐
6	Creek	☐	☐
7	Crickets	☐	☐
8	Owls	☐	☐
9	Frogs	☐	☐
10	Wolves	☐	☐
11	Wind on Grass	☐	☐
12	Wind	☐	☐
13	Fire	☐	☐

Overview about the sounds meanings:

3. **The sound of forest birds means the state of the traffic is normal behaviour.**
4. **Table 2 contains the expected seven malicious behaviours and their sounds.**

Table 2: The meaning of the sounds of the expected malicious behaviour

No	Activity name	The sounds will be played
1	**FIN scan**	The sound of "Cricket" will be heard tell us that we are receiving a high number of FIN packets then "Owl" sound will be heard telling us the number of incoming FIN packets are not because terminating normal flow connections showing that the number of incoming FIN packet is far higher than the number of the outgoing FIN packets plus the outgoing RST packets. This means our machine is receiving malicious FIN packet. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets
2	**SYN scan**	The sound of "Rain on Roof" will be heard tell us that we are receiving a high number of SYN packets then "Heavy Rain" sound will be heard telling us the number of incoming SYN packets is considered a bit higher than the outgoing SYN-ACK packets for accepting the connection handshake mechanism. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets. Because the number of SYN packets in this attack is high the sound of "Thunder" will he heard telling us that. The sound of "Wind on Grass" will be heard telling us the number of outgoing RST packets is high, telling us a high number of ports is being scanned.
3	**NULL scan**	The sound of "Frogs" will be heard tell us that we are receiving a high number of Null packets. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets. The sound of "Wind on Grass" will be heard telling us the number of outgoing RST packets is high, telling as a high number of ports is being scanned.
4	**Xmas ping scan**	The sound of "Wolves" will be heard tell us that we are receiving a high number of URG-PSH-FIN packets. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets.
5	**XMAS scan**	The sound of "Wolves" will be heard tell us that we are receiving a high number of URG-PSH-FIN packets. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets. The sound of "Wind on Grass" will be heard telling us the number of outgoing RST packets is high, telling us a high number of ports is being scanned.
6	**SYN DoS Flood**	The sound of "Rain on Roof" will be heard tell us that we are receiving a high number of SYN packets then "Heavy Rain" sound will be heard telling us the number of incoming SYN packets is considered a bit higher than the outgoing SYN-ACK packets for accepting the connection handshake mechanism. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets. Because the number of SYN packets in this attack is high the sound of "Thunder" will he heard telling us that. The sound of "Creek" will be heard telling us the number of outgoing SYN packets is very high, telling us this attack is DoS attack and not a scan. Also, the sound of "Fire" will be heard telling us that our machine is receiving a very high number of IP flow or Traffic flow in an unexpected way.
7	**DDoS using Spoofed IP's**	The sound of "Frogs" will be heard tell us that we are receiving a high number of Null packets. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets. Then the sound of "Fire" will be heard telling us that our machine is receiving a very high number of IP flow or Traffic flow in an unexpected way. This means we are receiving DDoS attack using null packets.

Appendix F

Training and guidance (sonification vs visualisation)

The training and guidance document for the study described in Chapter 5 is reproduced on the following pages.

Training and Guidelines Sheet (Snort) and (SoNSTAR)

This training and guideline sheet helps to learn how (Snort) and (SoNSTAR) works. It facilitates the learning of the sounds used in this system and to facilitate the distinction between them in order to confirm the activity received.
Please note that the experiment has three monitoring task. Snort then SoNSTAR then Snort and SoNSTAR together.

Monitoring Training Tasks 1 (Snort):

1. **Snort mostly relies on a "known bad" or "suspected bad" approach, observing traffic for patterns that correspond to malicious or suspicious activity. Snort is configured as when Snort detects one of the expected malicious activity, Snort will present the name of the malicious activity immediately on the screen for the period of the time window. We configured Snort to present the name of the attacks as used in this experiment and shown in Table 1.**
2. **We will perform one of the attacks and please If you see the activity name appeared on the screen, check the left Box for Yes in Table 1. If you changed your mind and would like to change your decision please check the right box for No and check the Yes box for the identified the activity name.**

Table 1: List of the expected malicious activities

Sl.	Activity name	Snort
1	ICMP ping	Yes ☐ No ☐
2	SYN Related port scan	Yes ☐ No ☐
3	FIN port scan	Yes ☐ No ☐
4	XMAS port scan	Yes ☐ No ☐
5	NULL port scan	Yes ☐ No ☐
6	SYN flood	Yes ☐ No ☐
7	DDoS or DoS spoofed IPs	Yes ☐ No ☐

Monitoring Training Tasks 2 (SoNSTAR):

3. **Table 2 shows the sounds which will be heard for each attack. Please keep this table with you when performing the monitoring task. Table 2 is will let you identify the activity name caused the set of heard sounds. Further understanding of the sounds is presented in Table 4.**
4. **Table 3 contains the sounds that are expected to be heard during the task. In this training, we will play sounds one by one. If you need us to repeat the sound, please ask. If you are satisfied with the audio learning please check the box in front of the sound in the Training 1 column.**
5. **We will demonstrate in real time one of the seven attacks used in the experiment, Please check the boxes provided for each sound heard in the Training 2 column. Then use these sounds to identify the activity name from the Table 2. During the task, you will be asked to identify the activity name based on the sounds heard in the same way used previously with Snort.**

Table 2: List of expected sounds

No	Activity name	The sounds will be played
1	**ICMP ping**	"Woodpecker"
2	**SYN Related port scan**	"Rain on Roof" and "Heavy Rain" and "Wind" and "Thunder" and "Wind on Grass"
3	**FIN port scan**	"Cricket" and "Owl" and "Wind"
4	**XMAS port scan**	"Wolves" and "Wind" and "Wind on Grass"
5	**NULL port scan**	"Frogs" and "Wind" and "Wind on Grass"
6	**SYN Flood**	'Rain on Roof" and "Heavy Rain" and "Wind" and "Thunder" and "Creek" "Fire" .
7	**DDoS using Spoofed IP's**	"Frogs" and "Wind" and"Fire"

Table 3: List of expected sounds

No.	Sound	Training 1	Training 2
8	Birds	☐	☐
9	Woodpecker	☐	☐
10	Rain	☐	☐
11	Rain on Roof	☐	☐
12	Heavy Rain	☐	☐
13	Rain and Thunder	☐	☐
14	Creek	☐	☐
15	Crickets	☐	☐
16	Owls	☐	☐
17	Frogs	☐	☐
18	Wolves	☐	☐
19	Wind on Grass	☐	☐
20	Wind	☐	☐
21	Fire	☐	☐

Overview about the sounds meanings for comprehension:

6. **The sound of forest birds means the behaviour of the traffic is normal.**
7. **Table 4 on the next page contains the expected seven malicious behaviours and their sounds.**

Table 4: The meaning of the sounds of the expected malicious behaviour

No	Activity name	The sounds will be played
1	**ICMP ping**	The sound of "Woodpecker" indicate receiving ICMP packets
2	**SYN Related port scan**	The sound of "Rain on Roof" will be heard tell us that we are receiving a high number of SYN packets then "Heavy Rain" sound will be heard telling us the number of incoming SYN packets is considered a bit higher than the outgoing SYN-ACK packets for accepting the connection handshake mechanism. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets. Because the number of SYN packets in this attack is high the sound of "Thunder" will he heard telling us that. The sound of "Wind on Grass" will be heard telling us the number of outgoing RST packets is high, telling us a high number of ports is being scanned.
3	**FIN port scan**	The sound of "Cricket" will be heard tell us that we are receiving a high number of FIN packets then "Owl" sound will be heard telling us the number of incoming FIN packets are not because terminating normal flow connections showing that the number of incoming FIN packet is far higher than the number of the outgoing FIN packets plus the outgoing RST packets. This means our machine is receiving malicious FIN packet. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets
4	**XMAS port scan**	The sound of "Wolves" will be heard tell us that we are receiving a high number of URG-PSH-FIN packets. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets. The sound of "Wind on Grass" will be heard telling us the number of outgoing RST packets is high, telling us a high number of ports is being scanned.
5	**NULL port scan**	The sound of "Frogs" will be heard tell us that we are receiving a high number of Null packets. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets. The sound of "Wind on Grass" will be heard telling us the number of outgoing RST packets is high, telling as a high number of ports is being scanned.
6	**SYN Flood**	The sound of "Rain on Roof" will be heard tell us that we are receiving a high number of SYN packets then "Heavy Rain" sound will be heard telling us the number of incoming SYN packets is considered a bit higher than the outgoing SYN-ACK packets for accepting the connection handshake mechanism. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets. Because the number of SYN packets in this attack is high the sound of "Thunder" will he heard telling us that. The sound of "Creek" will be heard telling us the number of outgoing SYN packets is very high, telling us this attack is DoS attack and not a scan. Also, the sound of "Fire" will be heard telling us that our machine is receiving a very high number of IP flow or Traffic flow in an unexpected way.
7	**DDoS using Spoofed IP's**	The sound of "Frogs" will be heard tell us that we are receiving a high number of Null packets. Then the sound of "Wind" will be heard telling us that our machine is sending out RST packets which means that scanned closed ports are replying with RST packets. Then the sound of "Fire" will be heard telling us that our machine is receiving a very high number of IP flow or Traffic flow in an unexpected way. This means we are receiving DDoS attack using null packets.

Appendix G

Publications

1. Vickers, P., Laing, C., Debashi, M., and Fairfax, T. 'Sonification Aesthetics and Listening for Network Situational Awareness'. *In SoniHED Conference on Sonification of Health and Environmental Data.* University of York, 12 September 2014.

2. Debashi, M. and Vickers, P. (2018). Sonification of network traffic flow for monitoring and situational awareness. *PloS One*, 13(4) (2018): e0195948.

3. Debashi, M. and Vickers, P. (2018). Sonification of Network Traffic for Detecting and Learning About Botnet Behavior. IEEE Access, 6(1), 33826–33839