Sonification of events generated by an SIEM

Luís Sousa and António Pinto

Abstract The information generated by a network monitoring system is overwhelming. Monitoring is imperative but very difficult to accomplish due to several reasons. In particular, there is some difficulty in the handling of so much real-time information in a way that is intelligible for the network administrator. Security Information Event Management applications, generate events that correlate multiple occurrences on the network. These events are classified accordingly to their risk. An application that allows the sonification of events generated by a Security Information Event Management, can facilitate the work of the network administrator by avoiding the necessity of him constantly monitoring the service and allowing him to just listen to the result of the sonification of such events.

Keywords: OSSIM, SIEM, Sonification.

1 Introdution

Monitoring a network to detect intrusions, vulnerabilities, and attacks is usually an arduous task. The manager of a network sometimes encounters a large amount of data which makes it difficult to task. There are several applications or open-source platforms that allow the network manager to have all the information he needs to monitor a network and detect attacks, such as Open Source Security Information Management (OSSIM) [3]. Was identified as the solution Open source and more complete in the study "Gestão de eventos de segurança de informação - SIEM" [5].

Luís Sousa
GCC, CIICESI, ESTGF, Polytechnic of Porto, Portugal, e-mail: 8090228@estg.ipp.pt

António Pinto
GCC, CIICESI, ESTGF, Polytechnic of Porto and INESC TEC, Porto, Portugal
e-mail: apinto@estg.ipp.pt

OSSIM allows the manager of a network some ease of monitoring because the main information about the state of the network and what is happening on the network is presented in the form of a dashboard where the main information is presented on a single screen. Even so the task of a network manager is not uncomplicated because it depends on the size of the network and above all requires continuous attention from those who are monitoring the network and using the OSSIM.

Recently, several researchers [16, 20], have been looking for alternative representations for network monitoring, like sonification techniques (data transformation into music) with several advantages.

The objective of this work is to create a solution to the problem that is evidenced, in other words, to create an application that allows sonification of the events that come from OSSIM. It makes it possible to simplify the work of those who are checking the status of the network, as there is no need to be consulting the service and only have to listen to the sonification results of such events. It allows focus on more important things while monitoring the network. With this it is possible to perform several tasks at the same time or to perform another task with more priority at that moment while getting a sense of the state of the network through the sound that is produced by the application.

This article is organized into sections, containing a total of 6 sections. Section 2 evidence in which the sonorization consists and the results of its application in several areas. Section 3 explains what SIEM is, what are its advantages. Section 4 presents the proposed solution to the problem in question. Section 5 speaks about the results of this solution. Section 6 focuses on a small conclusion on the subject and the article, and a future work is apresented.

2 Sonification

Sonification is a way of transforming data and relationships into an acoustic signal for interpretation and/or communication purposes [16].

According the article [9] which presents some definitions on the subject, sonification can only be called sonification if the sound is objective if its transformation is systematic and if it is reproducible, in other words, the sound results must be structurally identical for the same input.

The capabilities of human hearing are different from the skills of recognizing patterns in a visual way. Humans have a temporal resolution greater to hear than to see, and in this way, can have better performance with information overlapping in the auditory domain than in the visual [11].

Another advantage is that humans can become accustomed to sound patterns that continue to be susceptible to change even if these are subtle changes [11].

Sonification appears to be an appropriate and criterious solution for monitoring systems, since it allows visual attention to be focused on other tasks or other work to be realized [10, 11].

When the data to be monitored is presented visually in a very complex way, or we get a lot of data on a single screen, an auditory display provides a useful and sometimes a substitute supplement for a visual display [10]. The audio is excellent in guiding, or forwarding the listener to key data [10].
Exist attempts and previous studies of the application of sonorization in data coming from a network of computers, but each one with very different approaches both in obtaining the data to be processed and also the form of sonification the data in question.

In article [7], a monitoring system has been created that allows operators to identify excessive network traffic and spam, transforming network events into acoustic signals. This allows the system administrator to focus on more important things, while monitoring the network through the acoustic signals that are reproduced.

The authors of the article [18], have elaborated a system that sounds in real time a network. It allows to alert the administrators of the operations that are being carried out in the same one, in which, both the abnormal traffic and the normal one were sonorized.

In the Interactive Network Sonification (InteNtion) [8] project, the goal was to create an innovative approach to network traffic monitoring by adding a new dimension, the sound. Traffic was analyzed using the SharpPCap library, collecting traffic, that was then, parsed and trasformed in sound to help the administrator efficiently detect intruders on the network. There are other projects that have similar approaches like: Songs of cyberspace [6], Stetho [14], NeMos [15], NetSon [21], SonNet [20].

In the Songs of cyberspace [6] project, sonification techniques are used to examine the flow of data from the network. The sound system is used to support the entire surrounding environment and the decisions to be made at the moment.

The NeMos [15] project is a client-server Java application for monitoring a distributed system with sound. The server captures the data and the client produces the sound that is captured. His main promise was to complement a visual system.

The project NetSon [21], is a system that allows a large-scale organization to monitor metadata on a network in real time through sound. Due to the volume of data being analyzed every 24 hours in a large organization, only relevant aspects are considered and processed.

The SonNet project [20], was developed in Java. It captures packets on a network and transforms them into sound according to the information of each packet. The captured packets are sent via the Open Sound Control (OSC) protocol to an object written in the language Chuck [19, 12]. OSC is a protocol that offers flexibility and allows communication between computers, sound synthesizers and other multimedia devices [22, 23]. Communication is done

by OSC messages, where each message contains a destination path, in this case the Chuck object and a variable number of arguments, in this case with data about the captured packets. In the use of OSC messages, in addition to not having a defined number of arguments to pass, the format of the message is independent of the transport layer [24]. The Chuck object then created receives the OSC messages with information about each captured packet and creates real-time sounds. The Chuck language was used because it is an audio programming language that allows you to create sound and music in real time, it is free, open source and is available for Mac OS X, Windows and Linux.

3 Security Information Event Management

An SIEM is an application that allows you to generate events based on occurrences in the network. These events are classified according to the alert level and the danger they present to the networt. In 2012 there were about 85 SIEM applications, paid and free [1]. Companies are increasingly using SIEM solutions to enlarge security and monitor their networks [13]. An example is OSSIM [3].
The OSSIM, is a unified platform developed by AlienVault, free, open source and based on the Debian operating system [2]. OSSIM has four main components [2]:

1. The Sensor receives the logs from the network devices and stores them locally through the *rsyslog* service. The OSSIM agent through yours plugins, parses and normalizes each type of *log* and sends everything to the server.
2. The Server performs the essential functions of SIEM, like risk assessment, aggregation and correlation of events received by the Sensor and still stores those events in the Database.
3. The Framework provides the Web interface for system administration, binds and manages the components and security tools that compose the OSSIM.
4. The Database My Structured Query Language (MySQL), save the events and the system configuration.

The Figure 1 show shortly the OSSIM architecture.

OSSIM has various functionalities [2]:

- Collection and normalization of logs;
- Prioritization of events and risk assessment;
- Analysis and correlation of events;
- Generation of alarms and response actions;
- Vulnerability analysis, intrusion detection and network monitoring;

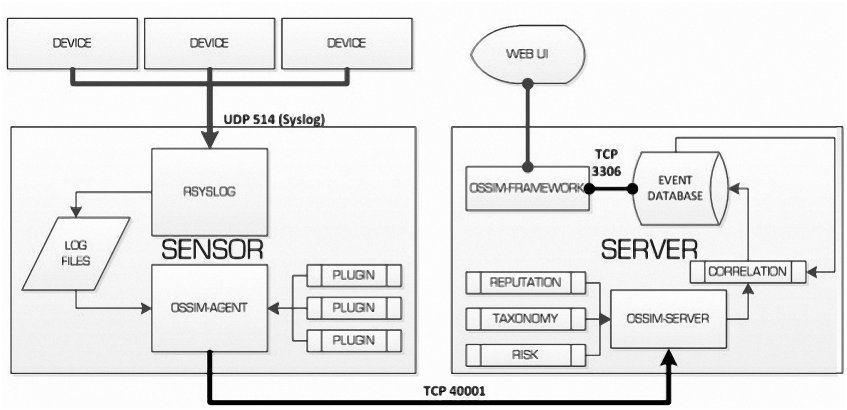

Fig. 1 OSSIM architecture [2].

In collection and normalization of logs, all events that are captured on the network, are saved and after analyzed and normalized, that is, are converted into a single format. In event prioritization and risk assessment, the server assigns priority values to the logged events. This allows know the danger/risk of a particular event, in order to alert the user. The risk of an event is calculated in real time using the following formula [4]:

$$risk = (value * priority * realibility)/25$$

The value refers to the importance level (between 0 and 5) of the machine that generated the event. It is manually assigned by who configured OSSIM. If it has not been assigned, it will default to 2. Priority refers to the importance of the event itself. It is a measure that is used to determine the impact an event might have on our network. It is between 0 (without priority) and 5 (high priority). Reliability is a value (between 0 and 10) inherent certainty an attack is real or not. OSSIM uses the value 0 for false positives and the value 10 for a real attack. As a result of the formula the risk can assume values between 0 and 10, such as: 0-2 (low), 3-4 (precaution), 5-6 (high), 7-8 (very high).

In the analysis and correlation of events, the events are analyzed and related to each other to detect possible attacks and anomalies.

In the generation of alarms and response actions, an event or group of events, under certain conditions, generate alarms. Alarms can be accessed from the OSSIM web admin panel. In addition, they can send *email* alerts to the system administrator or execute certain *scripts*.

In vulnerability analysis, intrusion detection and network monitoring, OSSIM has many free and popular open source tools:

- OpenVAS, is a powerful framework for the detection of vulnerabilities. It is considered the most advanced open source tool in the management and

search of vulnerabilities. Discover and scan for vulnerabilities in multiple hosts, concurrently.

- PRADS, is used to identify hosts and services, monitoring network traffic passively. Gathers information about hosts and services on the network.
- Nessus is a popular tool for detecting vulnerabilities in networked equipment. Supports search for vulnerabilities in an unlimited number of IPs, search for vulnerabilities in web applications, export reports, send e-mail notifications, and allow scheduled vulnerability scans at a particular time, among other features.
- Nmap is a cross-platform tool, used mostly to analyze a network and find open ports. It is a powerful, free and open source tool. It allows to identify hosts available on the network, which services are being used by these hosts, what operating systems they use, what type of firewall, what type of devices they are and what their MAC addresses are, among other features.
- Snort is NIDS/NIPS used essentially to detect intruders in the network. It is free software, with the ability to analyze network traffic in real time, analyze protocols and detect various attacks.
- Suricata is used to detect intruders on the network and is NIDS/NIPS by default of OSSIM 5 or up.
- TCPTrack makes it possible to monitor TCP connections. It is a software used to monitor network connections. Show real time IP traffic consumption, show client IP and source port, server IP and destination port, connection state ("established", "closed", etc), connection time, and the average data throughput on existing connections (bandwidth).
- Nagios is an open-source tool, popular in network monitoring. With this tool, it is possible to monitor hosts and their services (SMTP, POP3, HTTP, etc.), alerting the user when problems occur.
- NetFlow is used to display network traffic and busy bandwidth. It allows you to check the source IP address, destination IP address, protocol used, source port, destination port, service type, textit timestamps of the stream, number of bytes, total packets checked in the stream, packets per second, bits per second, average bits per second, and also the duration of the stream.
- Osiris is an HIDS, that periodically monitors one or more hosts. It maintains detailed information about changes to the systems, users, groups and kernel modules.
- OSSEC is another open-source HIDS, that can be used in OSSIM that also allows detection of intruders in hosts. Supports logs analysis, detection of system integrity and logs files, detection of rootkits, process monitoring, and a lot of other features.
- Snare is a set of agents that allow you to gather data from a wide variety of operating systems and applications, centralizing them and facilitating their analysis (logs).

The main advantage of OSSIM is the centralization of information in which it takes advantage of the diversity of the mentioned above tools and allows to make the registers of the same available in a concise and prioritized form in a single platform.

4 Proposed Solution

In order to allow the administrator to do other tasks while monitoring the network, the objective of this project is to develop an application to create acoustic signals for each network event generated by OSSIM.

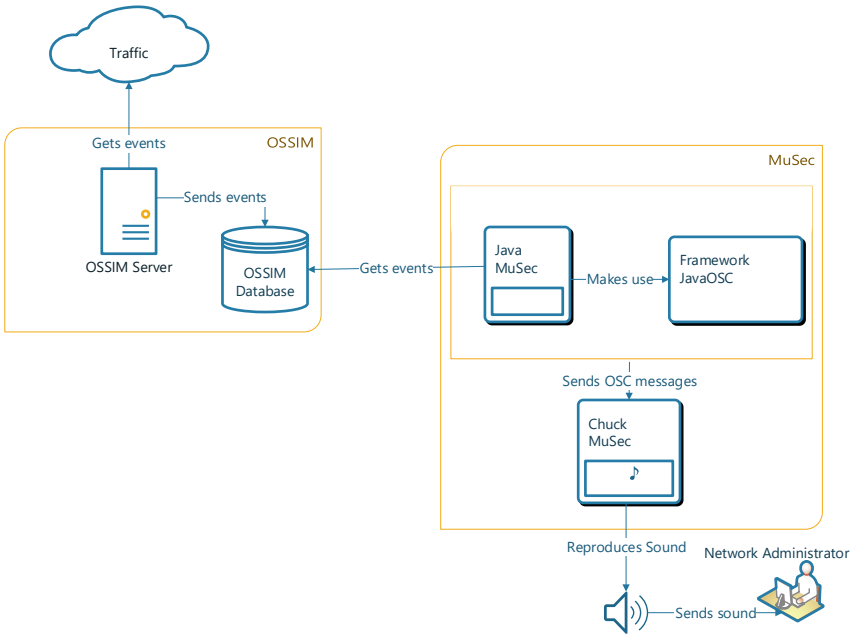

Fig. 2 Application architecture.

The application Music-enabled Security (MuSec), was written in the Java and Chuck languages. Is a simple and objective application, without additional configurations, that works in parallel with the OSSIM and takes full advantage of the hearing capacities of a human. The proposed solution consists of two components: Java MuSec e Chuck MuSec. Figure 2 presents an draw of the architecture for the proposed application.

The traffic is captured by OSSIM, which then does the internal processing of each captured event. It categorizes events by level of risk, priority, reliability, and other features, and stores information in a MySQL database, generating yours logs.

The Java MuSec accesses the OSSIM MySQL database and extracts useful information about each event, in other words, each packet captured. After the Java Musec component, with the help of the framework JavaOSC [17], communicates through the OSC protocol with an object written in Chuck language, the Chuck MuSec. In this communication, OSC messages are sent with information about a particular event, mainly its characteristics such as: (risk, value, priority and reliability) previously collected in the OSSIM database. Lastly, the Chuck MuSec, through the characteristics received, turns this event into acoustic signals that will be listened by the network administrator. Each risk level was mapped to a particular sound. These sounds are wav format musical loops that represent calm, relaxed sounds at low risk levels, or heavier sounds such as heavy metal loops and hard rock loops that represent high levels of risk. The sound discrepancy between these sounds allows you to efficiently alert the network administrator if something is affecting the network or not.

5 Results

The proposed solution allows to help the administrator to monitor the network with a set of functionalities:

1. **Alerts Collection:** The solution proposal allows collect events from an OSSIM server by accessing your MySQL database.
2. **Events Sonification:** The solution proposal allows sonifying the collected events.
3. **Cross-platform:** The solution proposal allows sonify events from an SIEM, independently of the operating system in use. It works on various operating systems such as Windows, Linux (see Figure 3) and Mac OS, and the network administrator can install it on your working computer, independently of operating system.
4. **Multi operating modes:** The solution proposal works through an execution without a graphical interface (command line) or through a graphical interface (JavaFX), allowing the user to choose the way he wants to interact with the application.
5. **Auto-executable:** The solution proposal works immediately on startup when deployed on a appropriate device. In this case the proposal was tested and successfully implemented in a Raspberry Pi 2.

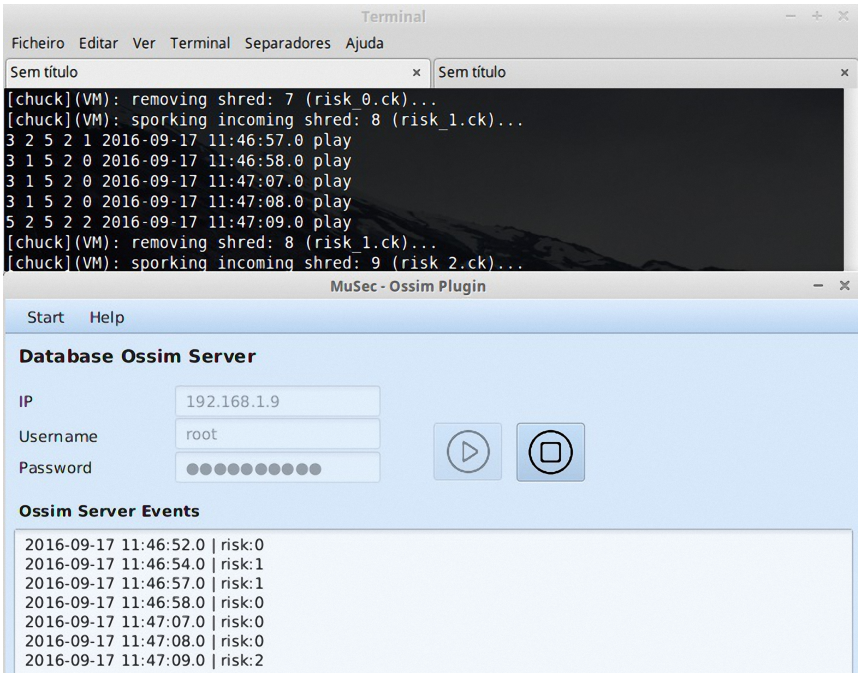

Fig. 3 MuSec working in operating system Mint 17.3 x86

6 Conclusion

So far there are many projects related with sonification techniques. They differ by the approaches used, the source of data to be sounded and the production of the sound of each project. Without hesitation, the origin of the data is something important in this context and must be something reliable, because the production of sound depends directly on the data obtained about the network. In this case, this task is instructed to a good tool and unified platform, the OSSIM. OSSIM tries to capture everything that goes on the network and to carefully qualify all traffic through levels of risk, levels of reliability, etc. Then it will only be necessary to transform this data from OSSIM into acoustic signals. If data reliability is something important in this type of projects, the sound production even more important is. The use of sound techniques depends essentially on how sound is produced, if it is audible and have criteria. The choice for sound production fell on the Chuck language because it is a free, open source language that allows real-time creation of sounds and music. OSSIM along with the Chuck language will certainly allow the administrator to do other tasks while listening to network monitoring to primarily detect attacks on it, but some improvements can be

assumed as future directions of research and application. It does not work on mobile platforms such as Android or iOS, compatibility of the MuSec application with other SIEM will allow to expand its use and finally the music variety that is currently produced by the MuSec application is not enough as initially wished.

References

1. M. Afzaal, C. Di Sarno, S. Dantonio, and L. Romano. An intrusion and fault tolerant forensic storage for a siem system. In *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*, pages 579–586, Nov 2012.
2. Marco Alamanni. Ossim: A careful, free and always available guardian for your network. *Linux J.*, 2014(242), June 2014.
3. Alienvault. Alienvault ossim: The world's most widely used open source siem, https://www.alienvault.com/products/ossim. Accessed: 2015-12-15.
4. USM AlienVault. Usm 5.1-5.2 asset management guide, rev.2. https://www.alienvault.com/doc-repo/usm/asset-management/AlienVault-USM-5.1-5.2-Asset-Management-Guide.pdf, 2015. Accessed: 2016-02-17.
5. João Alves. Gestão de eventos de segurança de informação siem. *Projeto Integrado, Licenciatura em Segurança Informática em Redes de Computadores, ESTGF, Politécnico do Porto*, nov 2015.
6. Mark Ballora, Nicklaus A Giacobe, and David L Hall. Songs of cyberspace: an update on sonifications of network traffic to support situational awareness. In *SPIE Defense, Security, and Sensing*, pages 80640P–80640P. International Society for Optics and Photonics, 2011.
7. Michael Gilfix and Alva L Couch. Peep (the network auralizer): Monitoring your network with sound. In *LISA*, pages 109–117, 2000.
8. Rudi Giot and Yohan Courbe. Intention–interactive network sonification. *Georgia Institute of Technology*, 2012.
9. Thomas Hermann. Taxonomy and definitions for sonification and auditory display. *International Community for Auditory Display*, 2008.
10. Thomas Hermann, Andy Hunt, and John G Neuhoff. *The sonification handbook*. Logos Verlag Berlin, GE, 2011.
11. Tobias Hildebrandt, Thomas Hermann, and Stefanie Rinderle-Ma. A sonification system for process monitoring as secondary task. In *Cognitive Infocommunications (CogInfoCom), 2014 5th IEEE Conference on*, pages 191–196. IEEE, 2014.
12. Ajay Kapur. *Programming for musicians and digital artists*. Manning Publ., 2015.
13. Alan Kebert, Bikramjit Banerjee, Glover George, Juan Solano, and Wanda Solano. Detecting distributed sql injection attacks in a eucalyptus cloud environment. In *Proceedings of the 12th International Conference on Security and Management (SAM-13), Las Vegas, NV, July*, 2013.
14. Masahiko Kimoto and Hiroyuki Ohno. Design and implementation of stetho—network sonification system. In *Proceedings of the 2002 International Computer Music Conference*, pages 273–279, 2002.
15. Delfina Malandrino, Daniela Mea, Alberto Negro, Giuseppina Palmieri, and Vittoro Scarano. Nemos: Network monitoring with sound. *Georgia Institute of Technology*, 2003.

16. Vincent F Mancuso, Eric T Greenlee, Gregory Funke, Allen Dukes, Lauren Menke, Rebecca Brown, and Brent Miller. Augmenting cyber defender performance and workload through sonified displays. *Procedia Manufacturing*, 3:5214–5221, 2015.
17. Illposed Software. Osc protocol library written in java, http://www.illposed.com/software/javaosc.html. Accessed: 2015-12-17.
18. Paul Vickers, Chris Laing, and Tom Fairfax. Sonification of a network's self-organized criticality. *arXiv preprint arXiv:1407.4705*, 2014.
19. Ge Wang. Chuck : Strongly-timed, concurrent, and on-the-fly music programming language , http://chuck.cs.princeton.edu. Accessed: 2015-12-17.
20. KatieAnna E Wolf and Rebecca Fiebrink. Sonnet: A code interface for sonifying computer network data. In *NIME'13—13th International Conference on New Interfaces for Musical Expression*, pages 503–506, 2013.
21. David Worrall. Realtime sonification and visualisation of network metadata. *International Conference on Auditory Display*, 2015.
22. Matthew Wright, Adrian Freed, Ahm Lee, Tim Madden, and Ali Momeni. Managing complexity with explicit mapping of gestures to sound control with osc. In *International Computer Music Conference*, pages 314–317. Citeseer, 2001.
23. Matthew Wright, Adrian Freed, and Ali Momeni. Opensound control: State of the art 2003. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, NIME '03, pages 153–160, Singapore, Singapore, 2003. National University of Singapore.
24. Woon Seung Yeo, Jonathan Berger, and Zune Lee. Sonart: A framework for data sonification, visualization and networked multimedia applications. In *Proceedings of the 2004 International Computer Music Conference*, pages 180–184, 2004.