MuSec: Sonification of alarms generated by a SIEM

Luís Sousa[1] and António Pinto[2]

[1] GCC, CIICESI, ESTG, Politécnico do Porto, Portugal
8090228@estgf.ipp.pt
[2] GCC, CIICESI, ESTG, Politécnico do Porto
and CRACS & INESC TEC, Porto, Portugal
apinto@inesctec.pt

Abstract. The information generated by a network monitoring system is overwhelming. Monitoring is imperative but very difficult to accomplish due to several reasons. More so for the case of non tech-savvy home users. Security Information Event Management applications generate alarms 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 security monitoring of a home network by a less tech-savvy user by allowing him to just listen to the result of the sonification of such events.

Keywords: Network security, 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. Open Source Security Information Management (OSSIM) [3] is an example that was identified as the preferable open source solution [5]. OSSIM allows the manager of a network some ease of monitoring because the main information about the state and what is happening on the network is presented in a dashboard. Even so the task of a network manager is not uncomplicated because it depends the size of the network and the utilization of OSSIM requires continuous attention and specific expertise in order to understand whats happening. A less tech-savvy home user is unable to use such a solution.
A possible approach is to convert the information produced by the SIEM into something more intelligible by the home user, such as sound or music. Recently, several researchers [15,**?**], 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 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.
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 presented.

2 Sonification

Sonification is a way of transforming data and relationships into an acoustic signal for interpretation or communication purposes [15]. In [9], the authors state that sonification can only be called sonification if the sound is objective, if the transformation is systematic and if is reproducible, in other words, the sound results must be structurally identical for the same input. The human hearing capabilities differ significantly from their visual capabilities. Humans have a greater temporal resolution of what they hear than of what they see, and in this way, can have better performance with overlapping information 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 performed while the persons go about their daily routine [10,11]. When a high volume of data to be monitored is presented visually 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].
Audio is excellent in guiding or forwarding the listener to key data [10]. There are some attempts and previous studies on the application of sonorization to data coming from a network of computers. Each one with a different approach with respect to obtaining the data to be processed and to the way the data is sonified. 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 [17], have elaborated a system that sonifies a network in real time. It alerts the administrators on operations that are being carried out in both the abnormal traffic and the normal one. 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 transformed 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 [13], NeMos [14], NetSon [20], SonNet [19]. 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 [14] 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. The authors main objective was to complement a visual system. The project NetSon [20] 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 [19], developed in Java, 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 Chuck language [18,**?**]. OSC is a protocol that offers flexibility and enables communications between computers, sound synthesizers and other multimedia devices [21,**?**]. Communication is done by OSC messages that do not have a predefined number of arguments and its format is independent of the transport layer [22]. A Chuck object 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 for the creation of 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 monitors a network and generates events based on occurrences in the network. These events are classified according to the danger they present to the network. In 2012 there were about 85 SIEM applications, paid and free [1]. Companies are embracing the use of SIEM solutions to enhance their network security and monitoring capabilities [12]. OSSIM [3] is an example application. It is an unified platform developed by AlienVault that is free, open source and based on the Debian operating system [2]. OSSIM has four main components [2]:

- **Sensor:** Receives the logs from network devices through the *rsyslog* service and stores them locally. After which, the OSSIM parses and normalizes each type of *log* and sends everything to the server.
- **Server:** Performs the risk assessment by aggregating and correlating the received events and by comparing them to a database know behaviors.
- **Web interface:** User for system administration, binds and manages the components and security tools that compose the OSSIM.
- **Database:** Stores the logs, events and the system configuration.

OSSIM includes functionalities [2] such as: the collection and normalization of logs; the prioritization of events and risk assessment; the analysis and correlation of events; the generation of alarms and response actions;

vulnerability analysis; intrusion detection and network monitoring. All captured events are saved, analyzed and normalized. In event prioritization and risk assessment, the server assigns priority values to the logged events. This server as a baseline for the establishment of the 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 of the machine that generated the event (values ranging from 0 to 5). This is manually assigned in the OSSIM configuration and has a default value of 2. Priority refers to the importance of the event itself. It is a measure that is used to determine the impact that the event might have on the network (values ranging from 0 to 5). Reliability is a value that indicates if an attack is real or not (values ranging from o to 10). OSSIM uses the value 0 for false positives and the value 10 for a real attack. All events are analyzed and correlated to each other to detect possible attacks and anomalies.

4 Proposed Solution

The proposed solution assume that the home user has an OSSIM set-up in his home network, where all his sensors and other devices are connected. Moreover, the OSSIM is assumed to be set-up to generate events in conformance to the users' expectation. The proposed solution must then satisfy the following requisites:

1. Collect events from an OSSIM server, accessing its database.
2. Sonify all collected events.
3. Operate independently of the underlying operating system (Windows, Linux or macOS).
4. Operate with or without a graphical interface.
5. Work immediately on startup when deployed on an appropriate device (such as a Raspberry PI).

The proposed solution, named Music-enabled Security (MuSec), creates acoustic signals for each network event generated by an OSSIM server. 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 architecture of the proposed solution, depicted in Figure 1, comprises two main components: the Java MuSec, and the Chuck MuSec. The network traffic is captured by OSSIM, which then does the internal processing of each captured event. It categorizes events by risk level, priority, reliability, and other features, and stores information in a MySQL database, generating logs.

The Java MuSec accesses the OSSIM MySQL database and extracts useful information about each event and, with the help of the framework JavaOSC [16], 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. After which, the Chuck MuSec, transforms the received events into acoustic signals

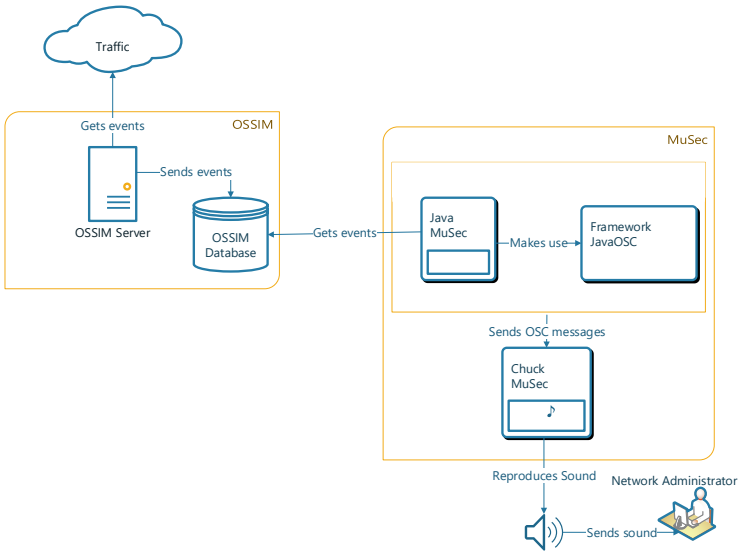

Fig. 1. Application architecture.

that will be listened by the user. Each risk level is mapped to a particular sound. The sounds are musical loops stored in wav files that represent relaxed sounds, when in the presence of low risk events, or heavier sounds such as heavy metal loops and hard rock loops, when in presence of high risk events. The discrepancy between these sounds efficiently alert the user when something is affecting the network.

5 Validation

To perform the validation of the proposed solution, functional tests were performed while using multiple operating systems. The goal was to verify if the application was running successfully in all supported operating systems. The application ran successfully in both command line and graphical modes in Linux Mint 17.3 x86 (see Figure 2), macOS X El Capitan x64, and in Windows 10 x64.

In the pursuit of the validation of the applicability of the proposed solution to a home scenario, the proposed solution was deployed on a Raspberry Pi 2, configured and setup to run in the command line mode at system startup. Afterwards, the (home) user would only need turn the device on listen to the event sonification. A OSSIM server was previously installed and setup to monitoring all network traffic. This setup was allowed to run for a one month period. Several scripts were developed to collect information about the state of the system and of the proposed solution. During this monitoring period, the Raspberry Pi 2 suffered multiple power failures but was able to automatically restart its sonification task.

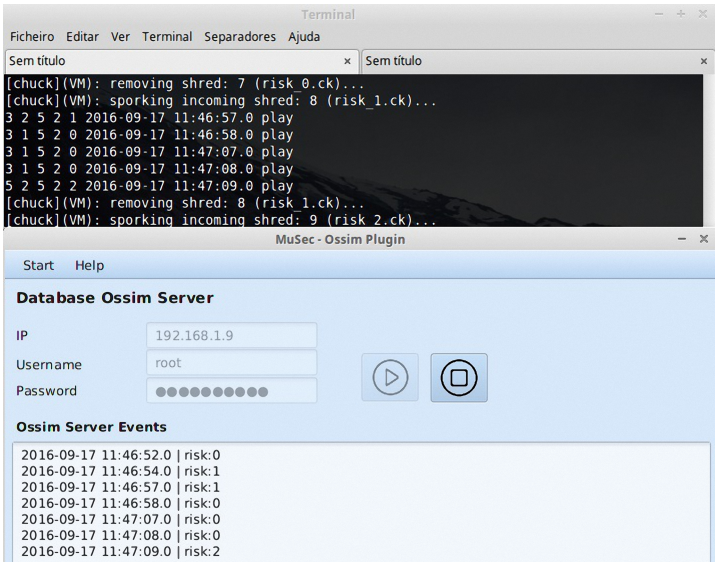

Fig. 2. MuSec running on a Mint 17.3 system with graphical interface

6 Conclusion

The proliferation of computers, smartphones and other devices that connect to home networks that, in turn, are connected to the Internet, is a reality in current days. On the one hand, the traditional home user does not understand or use security monitoring systems due to its complexity. On the other, attacks or security incidents that make use of any Internet connected device is recurrent. Most of the times, the home user is unaware of its participation. There are security monitoring solutions, some of which are even free to use and capable of detecting a significant number of attacks, such as the OSSIM, but the home user is unable to used on his daily life.

The proposed solution reduces this gap by allowing non tech-savvy home users to perform security incident monitoring without requiring any technical expertise or background. The proposed solution achieves this by using sound, through the sonification of the events generated by an OSSIM server. The home user has only to listen to the sound and, upon its change, will understand if his home network is being attacked or not.

References

1. Afzaal, M., Di Sarno, C., Dantonio, S., Romano, L.: 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. pp. 579–586 (Nov 2012)

2. Alamanni, M.: Ossim: A careful, free and always available guardian for your network. Linux J. 2014(242) (Jun 2014), http://dl.acm.org/citation.cfm?id=2642922.2642924
3. Alienvault: Alienvault ossim: The world's most widely used open source siem, https://www.alienvault.com/products/ossim, accessed: 2015-12-15
4. AlienVault, U.: 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. Alves, J.: Gestão de eventos de segurança de informação siem. Projeto Integrado, Licenciatura em Segurança Informática em Redes de Computadores, ESTGF, Politécnico do Porto (nov 2015), http://www.estgf.ipp.pt/ apinto/students/jalves_undergrad_2015.pdf
6. Ballora, M., Giacobe, N.A., Hall, D.L.: Songs of cyberspace: an update on sonifications of network traffic to support situational awareness. In: SPIE Defense, Security, and Sensing. pp. 80640P–80640P. International Society for Optics and Photonics (2011)
7. Gilfix, M., Couch, A.L.: Peep (the network auralizer): Monitoring your network with sound. In: LISA. pp. 109–117 (2000)
8. Giot, R., Courbe, Y.: Intention–interactive network sonification. Georgia Institute of Technology (2012)
9. Hermann, T.: Taxonomy and definitions for sonification and auditory display. International Community for Auditory Display (2008)
10. Hermann, T., Hunt, A., Neuhoff, J.G.: The sonification handbook. Logos Verlag Berlin, GE (2011)
11. Hildebrandt, T., Hermann, T., Rinderle-Ma, S.: A sonification system for process monitoring as secondary task. In: Cognitive Infocommunications (CogInfoCom), 2014 5th IEEE Conference on. pp. 191–196. IEEE (2014)
12. Kebert, A., Banerjee, B., George, G., Solano, J., Solano, W.: 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)
13. Kimoto, M., Ohno, H.: Design and implementation of stetho—network sonification system. In: Proceedings of the 2002 International Computer Music Conference. pp. 273–279 (2002)
14. Malandrino, D., Mea, D., Negro, A., Palmieri, G., Scarano, V.: Nemos: Network monitoring with sound. Georgia Institute of Technology (2003)
15. Mancuso, V.F., Greenlee, E.T., Funke, G., Dukes, A., Menke, L., Brown, R., Miller, B.: Augmenting cyber defender performance and workload through sonified displays. Procedia Manufacturing 3, 5214–5221 (2015)
16. Software, I.: Osc protocol library written in java, http://www.illposed.com/software/javaosc.html, accessed: 2015-12-17
17. Vickers, P., Laing, C., Fairfax, T.: Sonification of a network's self-organized criticality. arXiv preprint arXiv:1407.4705 (2014)

18. Wang, G.: Chuck : Strongly-timed, concurrent, and on-the-fly music programming language , http://chuck.cs.princeton.edu, accessed: 2015-12-17
19. Wolf, K.E., 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)
20. Worrall, D.: Realtime sonification and visualisation of network metadata. International Conference on Auditory Display (2015)
21. Wright, M., Freed, A., Lee, A., Madden, T., Momeni, A.: Managing complexity with explicit mapping of gestures to sound control with osc. In: International Computer Music Conference. pp. 314–317. Citeseer (2001)
22. Yeo, W.S., Berger, J., Lee, Z.: Sonart: A framework for data sonification, visualization and networked multimedia applications. In: Proceedings of the 2004 International Computer Music Conference. pp. 180–184 (2004)