SONIFICATION OF INFORMATION SECURITY EVENTS IN AUDITORY DISPLAY: TEXT VOCALIZATION, NAVIGATION, AND EVENT FLOW REPRESENTATION

Andrey Vishnevsky[1], Claudio Ruff Escobar[2], Marcelo Ruiz Toledo[3], Nadezhda Y. Abbas[4]

[1234] Cybersecurity center, University of Bernardo O'Higgins, Santiago, Chile

[1] ORCID: https://orcid.org/0000-0003-1451-8399 [2] ORCID: https://orcid.org/0000-0003-1954-0800

[3] ORCID: https://orcid.org/0000-0003-1865-7839 [4] ORCID: https://orcid.org/0000-0002-1667-1824

[1] andrey.s.vishnevsky@gmail.com, [2] cruff@ubo.cl

[3] mruiz@ubo.cl, [4] nadezda.abbas@ubo.cl

Received: 2021-09-12 | Accepted: 2022-05-05 | Published:2022-05-31

Abstract: This research is dedicated to developing an information security tool with a sound interface. If there is a possibility to manage information security by ear, analysis of computer attacks could be effectively maintained by people with vision problems. For this purpose, a human-computer interface is required in which the signs of malicious code and computer attacks are encoded using sounds. This research highlights the features returned by the console tools for static analysis of executable files, as well as an audio coding method for auditory expression of textual and non-textual features is proposed. In order to provide visually impaired people with the opportunity to work in the cybersecurity field, we propose a method of analysing malware by ear.

Keywords: audialization, malicious code, human-computer interaction, bisection.

Introduction

Information security is a huge research field due to ubiquitous digitalisation. One of the main risks of computer usage is threats related to malware spreading and network vulnerabilities. Organizations and individuals hold financial and reputational damage caused by various types of computer attacks, particularly by data leakage and ransomware infections.

The information security ecosystem includes antiviruses, firewalls, event management systems, vulnerability scanners for scanning files, network activity and evaluation of system behaviour. Vendors aim to automate information security products by implementing API and command-line interfaces which process textual data.

Nowadays, a lack of highly qualified specialists in the area of information security is observed (Fourie et al.,2014). On the one hand, this problem could be solved by involving college and university students, school pupils and other specialists without working experience (Mikhail et al., 2003). On the other, the involvement of disabled people could be a solution. However, this approach is used much less often (McDonnall & Crudden, 2015).

According to the World Health Organization, worldwide, in 2012, there were about 39 million blind people and 246 million people with poor eyesight (World Health Organization, 2012). And in 2020, there were 33.6 million of complete blindness cases globally (Vision Loss Expert Group of the Global Burden of Disease Study, 2020). Currently, they are unable to work professionally in information security due to vision problems. But using sonificated tools, helping to detect malware could resolve the problem and increase workplaces. In these consequences, using audialization methods and techniques might be a solution.

Sonification techniques are used for expressing data using sound. They are vastly applied in various areas, from biomedicine to seismology, and they are based on methods of acoustics, computer science, linguistics, and music theory (Dorigo et al., 2013). The features of computer attacks could be sonified by the sequences of musical sounds, noises, and speech recordings.

This is the reason why we consider that sonification is a key that will help visually impaired people work in cybersecurity.

This paper provides results of audialization of behavioural events, sources of computer attacks, sensor states of security sensors and the content language of the attack content.

In order to help the operator of an information security system to react efficiently to a computer attack we have encoded several features: the intensity of events, the danger degree, the novelty and source of the threat, the content language of the attack tools, the type of computer attack (malicious executable file or link to an infected website), the placement of elements in the list of features, the element number and the length of the signs list, arbitrary text strings. The composed sound interface is semantically significant for the operator of the information security system. In particular, the sensor state, the geographical location of the attack source, the attacker's language, and the tempo and type of attack were encoded by classical music fragments.

Subject area overview

Sonification techniques are vastly used to facilitate computer access for blind and visually disabled people. For this purpose, relief-point displays and means of speech output were created. In this case, the text is voiced using screen access programs and sound synthesizers (Dorigo et al. 2013). Qualitative features are sonified by musical instruments with different timbres, and quantitative ones - with sounds of different frequencies and loudness. Such non-verbal audio interfaces use the sound of real-world objects, familiar audio signals and surrounding sound (Keller & Stevens, 2004; Frauenberger & Noisternig, 2003).

Musical sounds are characterized by a period of oscillation, a linear spectrum, as well as volume, pitch, and timbre (Hermann et al., 2011; Kuznecov, 1989). A person could perceive by ear the melody, rhythm, tonality, and timbre of the instrument, and recognize music style, the composer, as well as the emotional colouring (Bożena, 2005). The perception of mode and completed musical phrases is observed in detail in paper (Noll & Clampitt, 2019). The group theory in music is described in depth in the works (Berry & Fiore, 2016; Hughes, 2015). The group consists of twelve notes of each octave according to the transposition operation (Peck, 2015).

Neuro linguists have found a parallel between the perception of speech and music: the human brain reacts to a melody ending on a false note (not corresponding to the rules of musical harmony) in the same way as it reacts to syntactic and semantic errors in sentences (Christiane, 2017). We propose to use these parallels between the perception of sounds and behaviour to form the basis of an information security tool with an audio interface.

Scientific studies related to sonification of complex data structures and information security events have appeared in recent years. In 2019, researchers demonstrated musical encoding of authentication events over nine continuous months from an enterprise network (Falk, 2019). They have transformed computer network traffic into music using volume-centric approach and number-based approach. In 2021, spider web architectures were translated into musical sounds (Su, 2021). Spider webs are similar to computer networks in terms of structure, so the proposed method can be used for the sonification of network security characteristics. In 2021, a musical interface for network security monitor based on high-level parameters of Baroque music was developed (Cai, 2021).

As a result of solving text to speech transformation tasks, IT giants have developed speech synthesisers for various languages. But some malicious codes may contain text lines not supported by the existing speech synthesisers, e.g. hexadecimal sequences or phrases in rare dialects. So, the sound interface of a computer attack detector should be able to vocalize arbitrary texts in a form convenient for the operator. Also, the operator may get tired after listening to a large amount of synthesised speech. That is why some audition load optimization is required.

To the best of our knowledge, there is no information security application which allows to investigate and block computer attacks solely by hearing. Academic researchers propose sonification methods for security data vocalisation but only as additional to the visual user interface. So, we are planning to fill the gap in the research field by developing an information

Volume 12, Issue 1. (CC) JACCES, 2022. ISSN: 2013-7087

security solution which combines text vocalisation with various sounds to exclude the necessity of visual work.

The implemented version of the audio interface uses Web Speech API and the BASS software library, as it is free and has sufficient functionality to change the volume, overlay sound recordings and work with various audio file formats.

Results

Auditory expression of textual features

We have studied the applicability of the existing speech synthesisers for developing sonified interfaces for malware scanners and network attack detectors. The resulting sonified applications rely on the free JavaScript library Web Speech API for vocalisation of English text and on our custom speech synthesiser based on the BASS library (Web Speech API; BASS Audio Library). The web application fetches verdicts of antiviruses via Virustotal API and vocalizes the downloaded report by speech or music according to the mode chosen by pressing keyboard buttons. The client part of our application is written on React JS, which acquires Node JS server for the sound recordings

The proposed sonified desktop application connects to a website that checks intercepted files and network requests using malware analysis tools. In order to extract features of executable file, the analytical subsystem firstly determined its type by its content, and then launched console reverse engineering utilities. The file type, compiler and installer version were determined by the Detect-It-Easy console program. The details of the digital signature contained in the file were determined by the Sysinternals Suite programs. The Yara scanner was used to search for signatures, and the Radare 2 toolkit was used to search for the characteristics of executable files. The analytical subsystem unpacked office documents using the OfficeMalScanner program so that the operator could check the presence of embedded malicious code.

At the top level of the menu, the operator is provided with a list of file or web page names to which the network requests have been registered. To go to the object features menu, the operator must press the “X” key. The keys “A”, “Z”, “Q”, “C”, “B”, and “M” are intended to scroll the list of objects, return to the top level of the list from the feature menu, and voice additional details depending on the type of observed object. These keys were chosen as their location is similar for the most popular keyboard layouts (QWERTY and COLEMAK).

Clicking “A”, “Z” keys in the features menu launch the analysis tools that extract the next or previous characteristics of the observed object. The output of the analysis tools is normalized to short text strings contained in the sound recording library. Table 1 contains an example of the normalization of the file type by the text output of the Detect-It-Easy utility. Firstly, the most significant feature is selected, then special characters, extra spaces and punctuation marks are removed from its text representation. After normalization, the string w remains; only this string should be voiced.

Table 1. Normalization of file type

№	Features returned by Detect-It-Easy	Normalized result
1	PE: compiler: Microsoft Visual C++(2013)[msvcrt]	
2	PE: library: Qt(5.6.3.0)[-]	library-pe-qt
3	PE: linker: Microsoft Linker(12.0)[EXE32,console]	

The collected features are played back by the audio recording via a JSON dictionary that contains a list of K keys as well as corresponding paths to audio files. String w corresponds to each normalized feature. If string w is entirely contained in the JSON dictionary of sound recordings, it would be played back. Otherwise, the substrings s_1,s_2…s_p, will be played back.

- **Step 1**. Initialize string w and list K.
- **Step 2**. Reset the countersi и j, initialize S as an empty set.
- **Step 3**. If $i<n$, go to Step 4, otherwise go to Step 9
- **Step 4**. If $j<|K|$, go to Step 5, otherwise go to Step 7
- **Step 5**. If substring $w_i \ldots w_{i+|k_j|}$ matches with k_j, add k_j to the set S.
- **Step 6**. Increment the counter j and go to Step 4
- **Step 7**. Append an empty character to each sequence from the set S.
- **Step 8**. Increment the counter i and go to Step 3
- **Step 9**. Select sequence $s_{(w,K)}$ with the smallest number of empty characters from the set S.
- **Step 10**. Return founded sequence $s_{(w,K)}$.

These substrings are used to decompose initial string according to the following algorithm, where i,j mean the character counter in the initial string and the counter in the list of K keys of the sound dictionary. The substrings $s_{(w,K)}$ for the resulting sequence are stored in the set S during splitting process of the initial string.

The string corresponding to the normalized feature will be played back only if it is contained completely in the dictionary of sound recordings. Otherwise, the player voices the substrings into which the initial string has been divided. The result of the implementation of the proposed algorithm is the splitting of the voiced string into the smallest possible number of substrings so that they contain the maximum number of characters of the initial string. In this case, the auditory representation of the string would be euphonious.

Navigating the feature list by ear

When launching the elaborated sonified information security tool, the operator hears the number of files that were collected. The operator can

Volume 12, Issue 1. (CC) JACCES, 2022. ISSN: 2013-7087

navigate through the list of files using the keyboard and play the list of extracted features. The proposed audio interface can sound the list of features line by line, or in larger parts to speed up the operator's work. The navigation process through the list of features by dividing it into equal parts is shown in Figure 1.

Figure 1 Sonification of features bisection

Bkav	tehtris
Lionic	ESET-NOD32
Elastic	Baidu
ClamAV	APEX
CMC	Avast
CAT-QuickHeal	Cynet
McAfee	Kaspersky
Malwarebytes	Alibaba

If dangerous features are not observed in the list operator will hear a special sound signal ⸙. If dangerous features are detected, the audio interface will represent them with sounds of different pitches so that each feature corresponds to one note of one of two chords of different pitches. The order of notes in ascending pitch corresponds to the order of features in ascending order of their numbers. For example, the combination of the first, second and fifth features corresponds to the consonance of the two lower sounds of the seventh chord in the small octave and the lower sound of the seventh chord in the first octave. The combination of the fourth and eighth features corresponds to the consonance of the upper sounds of both seventh chords.

Using the keyboard, the operator selects the number of parts p into which he is going to divide the list of features and the number of simultaneously sounding notes m that make up chords or consonances that he can distinguish. If the operator cannot distinguish not a single chord, the audio interface will reproduce the note of each feature separately. Otherwise, the audio interface will split the list of features into p parts, where $2 \le p \le 9, p \in N$, and then voice the placement of essential features in the resulting split using a chord, containing m notes.

Figure 1 displays the process of sonification of a list of 15 with only one dangerous feature at p=2. The audio interface splits the list of features into two parts. Then the first and second halves of the features are voiced in turn. The first half of the features is voiced by noise to show the absence of important data. Then the second half of the features is recursively divided into two equal parts, until the operator reaches the only one dangerous feature. The whole process requires the operator of the deceptive system to listen to 2·log_2^16=8 sound recordings, but does not require a musical ear, i.e. distinguishing notes. The time complexity of the searching process of an essential feature for a list of n elements is equal to

$$f(p) = p \cdot \lceil log_p n \rceil \qquad (1)$$

If the operator has at least a slightly developed musical ear, and he is able to distinguish the consonances of two notes m=2, then the search process could be reduced by half due to the encoding of pairs of features. Instead of listening to the sound recording corresponding to the first and then the second half of the signs step by step. In this case, the operator immediately hears the consonance corresponding to one of the four variants of the presence of essential signs in each of the parts. The four possible variants are encoded

Volume 12, Issue 1. (CC) JACCES, 2022. ISSN: 2013-7087

using the consonance D-d1,D,d1,⸙, where ⸙ denotes the absence of essential features.

The maximum number of consonances that the interface offers to remember is 256. These consonances include all combinations of notes from the 7th chord in the small octave and the seventh chord in the first octave. The time complexity when encoding combinations of features using consonances is equal to

$$\boldsymbol{f(m,p)} = \left\lceil \frac{\boldsymbol{p}}{\boldsymbol{m}} \right\rceil \cdot \left\lceil \boldsymbol{log_p\, n} \right\rceil \qquad (2)$$

Let us calculate the complexity of searching for a dangerous feature by the audio interface, depending on the number of simultaneously sounding notes from two seventh chords that the operator is able to distinguish.

Statement 1. For any fixed number of parts p, into which the list of features is divided, the minimum time complexity is achieved when the number of simultaneously sounding notes m is equal to p.

Proof. Let $m_1<p<m_2$. As $p,m_1,m_2 \in N$, for a smaller number of notes, the following inequality is true $\lceil p/m_1 \rceil \geq 1$, and for a larger number of notes, right $\lceil p/m_2 \rceil = 1$. Therefore,

$$\boldsymbol{f(m_1,p)} = \left\lceil \frac{\boldsymbol{p}}{\boldsymbol{m_1}} \right\rceil \cdot \left\lceil \boldsymbol{log_p\, n} \right\rceil > \left\lceil \boldsymbol{log_p\, n} \right\rceil = \boldsymbol{f(m_2,p)} \qquad (3)$$

Thus, for a fixed p and m<p, the time complexity function is equal to a constant less than any value of the same function for m>p, qed.

Statement 2. For any fixed number of simultaneously sounding notes m, the minimum time complexity is achieved when the number of parts p into which the list of features is divided is equal to m.

Proof. For any $p_1<m$, $p_1 \in N$ the following inequality is true

$$\boldsymbol{f(m,p_1)} = \left\lceil \frac{\boldsymbol{p_1}}{\boldsymbol{m}} \right\rceil \cdot \left\lceil \boldsymbol{log_{p_1} n} \right\rceil = \left\lceil \boldsymbol{log_{p_1} n} \right\rceil > \left\lceil \boldsymbol{log_m n} \right\rceil = \boldsymbol{f(m,m)} \qquad (4)$$

Consider the case when $p_2>m$, $p_2 \in N$.

$$\boldsymbol{f(m,p_2)} = \left\lceil \frac{\boldsymbol{p_2}}{\boldsymbol{m}} \right\rceil \cdot \left\lceil \frac{\boldsymbol{ln\, n}}{\boldsymbol{ln\, p_2}} \right\rceil > \frac{\boldsymbol{p_2}}{\boldsymbol{m}} \cdot \frac{\boldsymbol{ln\, n}}{\boldsymbol{ln\, p_2}} \qquad (5)$$

Volume 12, Issue 1. (CC) JACCES, 2022. ISSN: 2013-7087

$$f'_{p_2}(m, p_2) = \left[\frac{1}{m}\right] \cdot \left[\frac{\ln n}{\ln p_2}\right] - \left[\frac{p_2}{m}\right] \cdot \left[\frac{\ln n}{p_2(\ln p_2)^2}\right] \quad (6)$$

$$f'_{p_2}(m, p_2) > \frac{1}{m} \cdot \frac{\ln n}{\ln p_2} - \frac{\ln n}{m(\ln p_2)^2} > 0 \quad (7)$$

The time complexity increases monotonically when p_2>m, so

$$f(m, p_2) > f(m, m) \quad (8)$$

Therefore, the minimum value of the complexity function is achieved when p=m, qed.

Consequence 1. With an optimal choice of the number of sounding not es and the splitting power, the time complexity is equal to

$$f(m) = \lceil \log_m n \rceil \quad (9)$$

We could come to the conclusion that to navigate through the list of 32 elements, an operator who distinguishes only 2 notes will be able to detect the essential feature in 5 iterations, an operator who distinguishes 4 notes – in 3 iterations, and those with a better musical ear – in 2 iterations.

Event flow sonification

The proposed audio interface encodes previously known computer attacks by music and unknown computer attacks by noise. A certain plot of the musical fragment corresponds to the certain type of computer attack. For example, sending a file to a virtual user is expressed by a dramatic musical fragment in which one of the characters directly harms another. The transfer of a link is musically expressed by a plot in which one character harms another character indirectly.

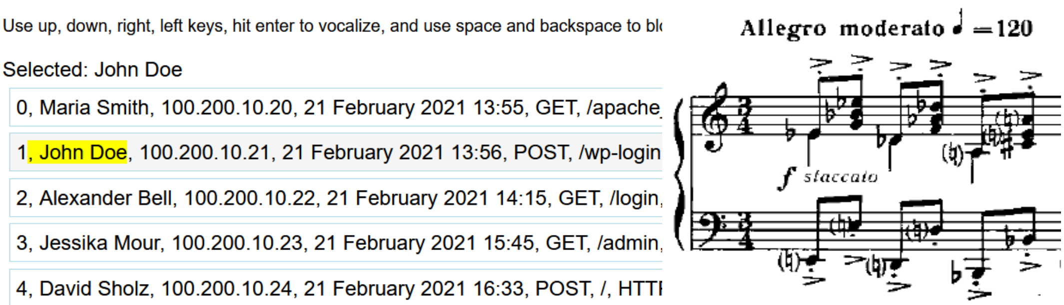
If it is possible to recognize the attacker's language, the audio interface will play back a vocal music characteristic for the country which is the origin of the computer attack. Excerpts from classical music were used for playing back recordings. The sounds of nature were divided into two groups: those that express dangerous and safe situations for humans and were grouped by noise intensity. Musical sound recordings also were grouped according to the plot, emotional colouring and tempo of the sound.

To simulate the attacking process and check the efficiency of the proposed audio interface, we have formed a list of sound recordings that express the flow of actions of the attackers who penetrated the website from the Commonwealth of Independent States (CIS) formed of nine post-Sovetic republics. It is presented in Table 1. It is assumed that the operator will associate the noise and music with the CIS region's nature or culture.

In order to express the intensity of the event flow, the sources of computer attacks, the harmfulness of the attacker’s actions and the novelty of the means of computer attacks, a collection of sound recordings were created. This collection includes sounds of nature from various world regions and fragments of musical compositions.

Musical compositions differ in tempo and time signature. Tempo is set by the number of beats per minute z and their duration. The alternation of down and up beats in a bar is determined by the size $(x¦y)$ – two natural numerals $x,y \in N$, where $x,y>1$, for example (2/4) – two quarters. The lower numeral of the size is a multiple of two and means the duration of the beat, which is uniquely determined by the tempo. The upper numeral indicates the number of beats per bar. For example, in Figure 2, the size is (3/4), and the tempo is 120 quarter fractions per minute.

Figure 2 An excerpt from the opera associated with a row of network attack description table



The time signature is recognized as simple if $x=2$ or $x=3$. If $x=2a$ or $x=3b$, where $a,b \in N$ time signature is recognized as complex. In all other cases, time signature is recognized as mixed (when $x=2a+3b$, where $x>1, x \in N$). There is one downbeat per one bar in simple time signature. In complex time signature there is $x/3$ downbeat per $x/2$ bar, depending if x multiple of 2 or 3. In mixed time signature, there are exactly $a+b$ downbeats.

The operator distinguishes by ear downbeats – strongest accents. The frequency of the downbeats could be calculated by the time signature and tempo of the music.

$$\boldsymbol{\nu}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \frac{\mathbf{1}}{\mathbf{60}\boldsymbol{z}} \cdot \frac{\boldsymbol{a}+\boldsymbol{b}}{\mathbf{2}\boldsymbol{a}+\mathbf{3}\boldsymbol{b}} \tag{10}$$

The number of downbeats, which will sound per a minute at a constant time signature and tempo of the music is equal to:

$$\boldsymbol{\lambda}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \frac{\mathbf{60}}{\boldsymbol{\nu}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})} = \boldsymbol{z} \cdot \frac{\boldsymbol{a}+\boldsymbol{b}}{\mathbf{2}\boldsymbol{a}+\mathbf{3}\boldsymbol{b}} \tag{11}$$

To correspond the music with a certain frequency of downbeats with attacker's behaviour, we have selected the timeline scale and defined each unit of the timeline during which the violator presented any action. The maximum frequency of the attacker´s actions within the scale of the timeline is considered the tempo of its behaviour. In order to measure it is necessary to define two attackers' actions separated by a minimum time interval Δt.

$$\boldsymbol{\lambda}_{\boldsymbol{behaviour}}(\Delta\boldsymbol{t}) = \mathbf{60} \cdot \frac{\mathbf{1}}{\Delta\boldsymbol{t}} \tag{12}$$

If the minimum time between the attacker's actions equals to 1 second, we will correspond the attacker's behaviour with music with frequency of downbeats equal to 1HZ or 60 downbeats per minute.

The emotional colouring of the music was used to encode the state of the sensors with which the attacker interacted. Melodies with national motifs corresponded to the geographical location of the sources of computer attacks.

Geospatial data helps to recognise the origins and targets of computer attack. Commonly advanced persistent threats are separated from regular computer attacks by country and nationality which victim belongs to. The novelty of the means of computer attack was encoded by applying noise to music. The more clearly the music is heard, the higher is the share of events that triggered signatures in comparison with the share of new unknown events that characterize the means of computer attack. Various noises help to express the intensity, harmfulness and source of each computer attack. For a flow of known events, we have identified two types of a tense state of honeypot: site hacking and transfer of a link or file. For each type of tense state we have determined a certain musical fragment.

Table 1 Sound recordings compared to attackers from CIS

Type of event flow	No suspicious actions	Detected hacking actions	Detected malware transferring
Weak flow of unknown events	Forest sound	Avalanche noise	Avalanche noise
A weak flow of know events	An overture from the ballet “Anyuta”	A scene from the ballet “Anyuta” before tarantella	The duel scene from the opera “Eugene Onegin “(without vocals)
A weak flow of known events and the language of attacker is known	Overture from the opera “Eugene Onegin”	Lisa’s aria from the opera “The Queen of Spades”	Tomsky’s aria from the opera” The Queen of Spades “
A strong flow of unknown events	Sound of wind in the forest	The sound of a blizzard	The sound of a blizzard
A strong flow of known events	Russian dance from the ballet “Swan Lake”	Odile scene from the ballet “Swan Lake”	Shostakovich’s Leningrad Symphony
A strong flow of known events and the language of attacker is known	Chorus of peasants from the opera “Eugene Onegin”	Aria of Tomsky from the opera “The Queen of Spades”	A riot scene from the opera “Boris Godunov”

Table 2 represents the names of musical fragments associated with attackers with the location in the Middle East.

Table 2 Sound recordings compared to attackers from the Middle East

Type of event flow	No suspicious actions	Detected hacking actions	Detected malware transferring
Weak flow of unknown events	Desert fauna sound	Bee flying	Bee flying

Type of event flow	No suspicious actions	Detected hacking actions	Detected malware transferring
A weak flow of know events	Unhurried cheerful Arabic music	Unhurried dramatic Arabic music	The scene of the murder of Vaclav from the ballet “The Fountain of Bakhchisarai”
A weak flow of known events and the language of the attacker is known	A slow cheerful Arabic song	Slow aria of the Shamakhan Queen from the opera “The Golden Cockerel”	Aria of the Turkish Sultan after the castle storming from the opera “Mahomet the Second”
A strong flow of unknown events	The sound horse galloping leisurely	Sandstorm sound	Sandstorm sound
A strong flow of known events	Fast fun Arabic music	Polovtsian dances from the opera “Prince Igor”	The scene of the palace capture by the Tatars from the ballet “The Fountain of Bakhchisarai”
A strong flow of known events and the language of attacker is known	Fast fun Arabic song	A quick aria of the Shamakhan Queen from the opera “The Golden Cockerel”	The scene of the capture of a Greek city from the opera “Mahomet the Second”

Discussion

The sound interface for information security solutions is still in the development stage. Therefore, the most frequent questions are related to the properties of music and the formal model used to design the auditory display. So, we are planning to use all easy to perceive characteristics of music like intonation and tempo. Also, we have proposed a mathematical approach base on game-theoretical modelling to encode information security events into musical pieces (Vishnevsky, 2022).

The usage of music volume was recognized as ineffective for continuous data signal where minor changes are significant (Falk, 2019). The macrostructure and high-level patterns of music are much more easily to identify for non-musicians than the pitches (Cai, 2021). So it makes sense to develop sound interfaces adaptable to the musical ear of the user.

Conclusions

In this study we have proposed and implemented a method of encoding information security events using plots of musical pieces, a method of verdicts lists musical sonification, and an algorithm for arbitrary text synthesis. Our initial prototype was written in C++, but we are redeveloping our sonified application in JavaScript to make it possible to work on various types of electronic devices. The created prototype opens the possibility of professional work for the visually impaired in the field of information security and adapts the complexity of detecting computer attacks to the musical ear of the security operator.

References

[1] Fourie L., Pang S., Kingston T., Hettema H., Watters P., Sarrafzadeh H. (2014). The global cyber security workforce – an ongoing human capital crisis. Paper presented at the 2014 Global Business and Technology Association Conference (pp. 173-184). Baku, Azerbaijan: Global Business and Technology Association.

[2] Mikhail M.B., Walther B.R., Willis R.H. (2003). The effect of experience on security analyst underreaction. 35(1), 101-116. Journal of Accounting and Economics. https://doi.org/ 10.1016/S0165-4101(02)00099-X

[3] McDonnall M.C., Crudden A. (2015). Building Relationships with Businesses: Recommendations from Employers concerning Persons Who Are Blind/visually impaired. 81(3), 43-70. Journal of Rehabilitation.

[4] World Health Organization. (2012). WHO releases new global estimates on visual impairment. Retrieved from http://www.emro.who.int/control-and-preventions-of-blindness-and-deafness/announcements/global-estimates-on-visual-impairment.html (accessed: 10.04.2022)

[5] Vision Loss Expert Group of the Global Burden of Disease Study. Causes of blindness and vision impairment in 2020 and trends over 30 years: evaluating the prevalence of avoidable blindness in relation to “VISION 2020: the Right to Sight”. Lancet Global Health 2020.. . https://doi.org/10.1016/S2214-109X(20)30489-7

[6] Dorigo M.L., Harriehausen-Mühlbauer B., Stengel I., Haskell-Dowland P.S. (2013). Nonvisual Presentation and Navigation within the Structure of Digital Text-Documents on Mobile Devices. Paper presented at the Universal Access in Human-Computer Interaction. Applications and Services for Quality of Life (pp. 311-320). Berlin, Heidelberg: Springer. . https://doi.org/10.1007/978-3-642-39194-1_37

[7] Keller P, Stevens C. (2004). Meaning from environmental sounds: types of signal-referent relations and their effect on recognizing auditory icons. 10(1), 3-12. Journal of Experimental Psychology: Applied. https://doi.org/10.1037/1076-898X.10.1.3

[8] Frauenberger C., Noisternig M. (2003). 3D Audio Interfaces for the Blind. Paper presented at the 2003 International Conference on Auditory Display (pp. 280-283). Boston, MA, USA: Boston University Publications.

[9] Hermann T, Hunt A, Neuhoff J.G (2011). The sonification hand-book. Berlin, Germany: Logos. isbn: 978-3-8325-2819-5

[10] Kuznecov L.A. Akustika muzykal’nyh instrumentov (1989). Spravochnik. Moskva, RSFSR: Legprombytizdat. isbn: 5-7088-0166-2

[11] Bożena K. (2005). Perception-Based Data Processing in Acoustics. Applications to Music Information Retrieval and Psychophysiology. Berlin, Heidelberg: Springer. isbn: 978-3-540-25729-5

[12] Noll T., Clampitt D. Exploring the Syntonic Side of Major-Minor Tonality (2019). Paper presented at the Mathematics and Computation in Music (pp. 125-136). Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-030-21392-3_10

[13] Berry C., Fiore T. (2016). Hexatonic Systems and Dual Groups in Mathematical Music Theory. 11(2), 253-270. Involve: a Journal of Mathematics. https://doi.org/10.2140/involve.2018.11.253

[14] Hughes J.R. Using Fundamental Groups and Groupoids of Chord Spaces to Model Voice Leading (2015). Paper presented at the Mathematics and Computation in Music (pp. 267-278). Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-319-20603-5_28

[15] Peck R.W. All-Interval Structures (2015). Paper presented at the Mathematics and Computation in Music (pp. 279-290). Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-319-20603-5_29

[16] Christiane N. (2017). Methods in Neuromusicology: Principles, Trends, Examples and the Pros and Cons (Schneider A., Ed.) Studies in Musical Acoustics and Psychoacoustics (341-374). Cham, Switzerland: Springer International Publishing. https://doi.org/10.1007/978-3-319-47292-8

[17] Falk, C. (2019) The 25th International Conference on Auditory Display (ICAD 2019). Sonification with music for cybersecurity situational awareness (50-55). Northumbria University, Newcastle upon Tyne, UK. https://doi.org/10.21785/icad2019.014

[18] Su I. (2021) Interactive exploration of a hierarchical spider web structure with sound. Journal on Multimodal User Interfaces. https://doi.org/10.1007/s12193-021-00375-x

[19] Cai Y. (2021) Compositional Sonification of Cybersecurity Data in a Baroque Style. In: Ahram T.Z., Karwowski W., Kalra J. (eds) Advances in Artificial Intelligence, Software and Systems Engineering. AHFE 2021. Lecture Notes in Networks and Systems, vol 271. Springer, Cham. https://doi.org/10.1007/978-3-030-80624-8_38

[20] Web Speech API https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API (accessed: 10.04.2022)

[21] BASS Audio Library https://www.un4seen.com/ (accessed: 08.08.2021)

[22] Vishnevsky A. (2022). Sonification of Information Security Incidents In An Organization Using A Multistep Cooperative Game Model. In: A. Rocha et al. (Eds.) WorldCIST: World Conference on Information Systems and Technologies 2022. Lecture Notes in Networks and Systems, vol 468, pp. 1-9. Springer Nature Switzerland AG 2022. https://doi.org/10.1007/978-3-031-04826-5_30

How to cite this article:

Vishnevsky, A., Ruff-Escobar, C., Ruiz-Toledo, M., & Abbas, N.Y., (2022). Sonification of information security events in auditory display: text vocalization, navigation, and event flow representation, 12(1), 116-133. Journal of Accessibility and Design for All. https://doi.org/10.17411/jacces.v12i1.359