Automated Measuring of Information Security Related Habits

Tobias Fertig, Andreas E. Schütz, and Kristin Weber
Faculty of Computer Science and Business Information Systems
University of Applied Sciences Würzburg-Schweinfurt
Sanderheinrichsleitenweg 20, 97074 Würzburg, Germany
{tobias.fertig, andreas.schuetz, kristin.weber}@fhws.de

Abstract

Since the digital age requires interaction with digital services, the information security awareness (ISA) of everyone gets more important than ever. Since the ISA is defined as a set of aspects, it is not enough to increase the knowledge. This work focuses on the aspect of habits. Therefore, we used design science research to create an artifact which allows the automated measurement of habits. The automation can be achieved through a client-server application which tracks the behavior of employees in a GDPR-compliant way and calculates multiple metrics based on the tracked behavior. However, not all of the defined metrics are applicable in every company. Therefore, additional process iterations of the design science research methodology are required.

1. Introduction

The digital age requires increasing interaction with digital services. Everyday life confronts people with online banking, online shopping, or the use of e-government services. The protective measures brought by the COVID-19 pandemic have intensified this development. Also, in home office settings, the working life is increasingly shaped by the use of the employer's IT infrastructure. In addition to the convenience that these digital processes bring with them, they also increase the requirements for information security. The manipulation, misuse, compromising, and willful destruction of data can have serious legal and economic consequences for companies. Technical hurdles make it more and more difficult to attack the technical infrastructure. Therefore, the human user is increasingly becoming the target of hackers and is thus an indispensable support for the defense of sensitive information. Users are confronted with attacks such as phishing, malware infected e-mails, or social engineering. On the other site, the study by [1] proves that people in a certain context contribute more reliably to information security than technical measures.

In order to prepare employees for their important role, they have to be aware of information security. The term information security awareness (ISA) describes the individual level of awareness of a person [2]. A person's ISA is made up of four factors: knowledge or skills, behavioral intention, salience, and habit [2]. The importance of habit for the behavior of a person in accordance with information security has so far received little attention in research. The factor has the side effect that with a strong habit, the importance of the influence of the behavioral intention of a person diminishes [3].

In order to give habit the necessary status when planning measures to increase the ISA, we have to measure the current degree of development of a habit for a certain behavior. In practice, methods such as the Self-Report Habit Index (SRHI) are used for this purpose. However, the SRHI only relies on the answers of test subjects themselves. In order to get a reliable statement about the degree of the habit of a person, especially with digital behavior, the support by an automated measurement is a promising option. The digital behavior of a user can then be monitored and evaluated. This paper presents the first iteration of a design science research process for the development of a software for automated habit measurement. The software has to be GDPR-compliant in order to record and evaluate a user's habits for certain behaviors within companies [4].

First, we will explain the theoretical background for our research regarding ISA, habits, and the measuring of habits in general. Second, we will present our used methodology and define our research questions. Afterwards, we will present our design as well as the implementation of our software tool. Moreover, we will answer our research questions and discuss the results. Finally we will give a short summary and an outlook for future work.

URI: https://hdl.handle.net/10125/80267
978-0-9981331-5-7

2. Theoretical Background

2.1. Information Security Awareness

ISA addresses the *human factor* and how users can be sensitized to show information security-compliant behavior. Targeting users requires less effort for attackers than technically attacking IT systems by using methods such as brute-forcing. Therefore, the active involvement of users is important for the information security concept of a company. One commonly accepted method for this involvement is an ISA campaign. Campaigns aim for motivating users to use their theoretical knowledge about information security in practice [5] and for convincing them of the importance of their actions.

To explain human behavior we use the Integrated Behavioral Model (IBM) [6]. The IBM was already interpreted in the context of ISA and used to explain the mental construct ISA [2]. Based on that, we used the IBM to interpret the findings of our analysis and draw initial conclusions for suitable ISA measures. The ISA of a person is the sum of the four factors knowledge and skills ("I know how the behavior is performed"), habit ("I'm used to perform the behavior"), salience ("The performance of the behavior is in my mind"), and behavioral intention ("I want to perform the behavior"). Especially the factor behavioral intention is complex and formed by the three mental constructs attitude, perceived norm, and personal agency of a person. These, in turn, are formed by emotions and beliefs.

The four indicated factors must be influenced to convince a user to behave compliant with information security policies. But even then, environmental constraints can still prevent the performance of the behavior. This shows that a person's environment is also significantly involved in shaping their behavior. In addition, the influence of environmental factors can also affect the behavioral intention [6]. In companies these environmental factors are shaped by the organization (e.g., the existence of policies, usability of services). This is also confirmed for ISA by [7], who name *organization* as an additional aspect of security awareness. The organization ensures that employees are able to behave in compliance with information security, i.e., no barriers exist, which are in conflict with compliant behavior. For example, a barrier is a password change link which is hidden in the depths of the intranet. At the same time, organizational measures, such as increasing the usability of an application, may support information security.

2.2. Habits

A habit is a learned sequence of actions that have become automatic responses to certain triggers [8]. Habits usually have the following characteristics: They are activated unconsciously, are difficult to control and are mentally efficient, so that other activities can be carried out in parallel [9].

The more often a behavior is carried out, the stronger the habit becomes [3]. For the establishment of a habit, however, it is important that satisfactory consequences result from the implementation of the behavior [10]. The emergence of a habit is usually the result of an initially intentional act [10], but from time to time this act is no longer carried out due to the motivation to achieve a goal. Rather, it is carried out by cues and henceforth cause the execution [11]. A distinction is made between "specific times, locations, moods and interaction partners" as four different groups of cues [12].

With increasing habit, the influence of the behavioral intention on the execution of the behavior decreases [3]. Because of the lesser influence of the factors that make up the behavioral intention, the process of executing a behavior is less complex. Locking the computer when leaving the work place is an example of a common activity. After it has been executed a few times, the employee will unconsciously do this as soon as they leave the workplace without worrying about the execution. So the behavior became a habit. However, habit is not equally relevant for all behaviors. Behaviors that are performed less often, such as reporting a security incident, do not simply become a habit due to the lack of repetition. If the employee notices a security incident, they will probably first think about what to do.

The strength of a habit is usually measured by how often a behavior was carried out in the past [12]. The most commonly used approach to measuring habit strength is the Self-Report Habit Index (SRHI) [13] from [14]. The SRHI consists of a qualitative survey. The test object evaluates twelve statements using a five- or multi-point Likert scale, with the value 1 for "I disagree" and the value 5 for "I agree". Questions are asked about the regular repetition of behavior in the past, unconscious triggering or identification with the corresponding behavior. The result is then calculated in relation to the total number of points. According to [15] a value below 25% is considered to be a low habit strength, from 26% to 82% as a medium habit strength and above 83% as a high strength of habit.

With its qualitative character the SRHI can help to generate hypotheses about habits regarding the information security related behavior of a person.

However, the hypotheses are based on the statements of the test subjects. According to [16] the social desirability bias (SDB) is especially critical [17] regarding ISA. Fertig et al. state that the subjects could give answers which are more favorably viewed by others instead of the truth. The hypotheses generated with the SRHI can be validated with an additional automated measuring system. Thereby discrepancies to the actual behavior can be detected. The actual measurement of the execution of a behavior corresponds to the definition of Ji und Wood [12].

3. Methodology and Research Questions

Our research is based on the Design Science Research Methodology from Peffers et al. [18]. Therefore, we identified a problem first. We conducted two systematic literature reviews and interviewed a total of six experts of the industry to gather the requirements [reference omitted due to blind review]. The literature revealed a lack of information regarding the measuring of ISA as well as the considering of habits regarding ISA of employees. Moreover, the interviews with the experts revealed that one possible reason for the lack, is the challenge in the quantification of human behavior. This is why often only metrics about the number of trained employees are used for ISA [19].

The problem identification revealed requirements for metrics which are identical in literature as well as in practice: metrics should be meaningful, practice-oriented, fast and easy to handle, as well as individual. However, the experts added additional requirements: automation, anonymity, and extensive documentation [reference omitted due to blind review].

Therefore, we derived the following objectives for our artifact: Our artifact has to allow us the definition of metrics which fulfill the aforementioned criteria. Moreover, the artifact has to gather the metrics automatically in order to allow them to be fast and easy to handle. The anonymity is required to be GDPR-compliant and is, therefore, also an objective for our artifact.

The requirements and objectives have been translated into the following research questions:

RQ1) How can habits be measured in the context of ISA?

RQ2) Can the measuring of habits be automated?

RQ3) If an automation is possible, can it be GDPR-compliant?

4. Concept for Automated Measuring

Since literature shows the use of SRHI for manual measuring of habits, we analyzed the default questionnaire to check for possibilities for automation. The items *... I do frequently*, *... that belongs to my (daily, weekly, monthly) routine*, and *... I have been doing for a long time* are focusing on the history of behavioral repetition. Since our software will track the behavior of employees, those items of the SRHI are possible to automate. Therefore, the results of the SRHI questionnaire could easily be compared to the results of the tracking tool.

This section summarizes the design of our artifact according to the methodology of Peffers et al. [18]. At first, we will define some metrics which can be used by our software to measure habits. Then, we will explain the derivable habits which our artifact will measure automatically. Finally, we will discuss how random behavioral patterns can be detected by our measuring tool.

4.1. Metrics for Automated Measuring

In this section, we describe our defined metrics. They are summarized in Table 1. We also provide the references as well as decisions which led to each metric. Since it is not allowed to track the complete behavior of an employee, we reduced the logging to only the security-relevant behaviors.

Using the same password over a longer period of time involves certain risks. Therefore, some institutions rely on expiring passwords, which must be changed after a certain number of days. However, this leads to consecutive passwords, which contain, for example, a number that is incremented with each forced change. It is also not uncommon for this measure to simply add the current month to the end of the password. However, to recognize passwords that are too old and to determine the frequency of password changes, the timestamps of the latest password change should be recorded [20].

Removable storage devices pose a massive threat to companies [21]. Removable storage devices include simple usb devices, external hard drives and memory cards, but also CD-ROMs or floppy disks. As part of a study, employees from the University of Illinois deposited 297 usb devices on their university campus. The first usb device was connected to a computer in less than six minutes. The experimental setup of the study led to a success rate of up to 98% [22]. Had this attack been carried out in a real scenario by a malicious attacker, this would result in fatal consequences. Usb devices can act as HID interfaces

with appropriate modifications. A usb devices pretends to be a keyboard, which simulates user input through previous programming. This opens a command line in a matter of seconds and can, for example, open a TCP connection to the attacker's server. The attacker receives full control over the computer. From there they can spread their attack on the local network. If the attacker has already advanced to this stage of his attack, they can now steal sensitive data or cause massive damage. Therefore, monitoring the connection of the devices to the computers is advisable. If a removable medium is connected contrary to the policies, this must be logged [23].

In addition to the monitoring of removable storage devices, the data transfer should also be monitored. A well-known reason is the example of the whistleblower Edward Snowden. He stole security-critical documents unnoticed from an NSA facility in Hawaii with the help of a usb device [24]. Therefore, it should be logged if sensitive data is transferred to removable storage devices.

Since drive-by downloads can happen during web browsing, an unconscious infection of systems with malware can happen. In addition, corporate data can get into private hands via offers such as web mail, cloud-based file sharing services, or FTP servers. As a consequence, it is advisable to record improper access to blacklisted domains or to potentially dangerous domains [20, 25].

If employees in companies do not lock their screen when leaving the workplace, this behavior enables unauthorized third parties to cause damage from the employee's computer. This primarily harms the company itself and ultimately falls back on the employee as it gives the impression that they caused the damage from their computer. Therefore, employees should always lock their screen when they leave their workplace. Even the short period of time it takes to get coffee is enough for an attacker to cause damage from the unlocked computer. For this reason, it makes sense that the idle time of the user is also logged, i.e., the time in which no keyboard inputs or mouse movements are made. If a user does not enter any data for a predefined period of time, it can be assumed that they have left his workplace. If this is the case and the computer has not been locked, this can be assessed as a policy violation which must be logged [20].

The most widely used operating system in offices, Windows 10, receives two new versions per year [26]. If an employee prevents operating system updates they present themselves to potential attackers as a simple target due to the contained and known uncertainties. From this it can be concluded that our tool should check the current Windows version against the supported version. Any mismatch must be logged.

The software installed on a computer and their respective versions have an impact on the security of a computer. In particular, common applications are often the focus of attackers, as they offer a simple attack vector. For example, program codes can be executed via prepared PDF documents in Adobe Reader versions that have not yet been patched [27]. Therefore, all program installations must be verified, the installation of prohibited software or the use of a program version that is no longer supported must be logged [20].

4.2. Derivable Habits

The characteristics of habits queried in the SRHI are: the regular repetition of behavior in the past; the difficult controllability; the unconscious release; and mental performance efficiency. First of all, it is now crucial which behavior patterns can be measured with regard to their habitualization.

Compliance with password regulations initially seems difficult to measure in terms of habit. It is obvious that different companies handle the composition of the password from upper and lower case letters, numbers and special characters differently. In addition, it is also a matter of the fact that passwords should not be passed on when absent - not even to your own colleagues. However, should this happen contrary to the password policies, this will probably rarely be the case. A check of this behavior with the SRHI can produce rather useless results. Moreover, since the change of passwords will not happen very often it cannot be defined as a habit. Therefore, we can measure the metric for password changes but cannot derive a valid habit from it.

The removal of suspicious e-mails would initially also be a behavior which, due to the irregular execution, is rather difficult to habitualize. However, a distinction must be made between the process of optical analysis of the e-mails by a user and the actual removal. Not every e-mail has to be removed automatically - but every incoming e-mail should first be checked visually for indicators of malicious content or attachments or phishing. The removal or reporting of potentially harmful e-mails is then only the follow-up activity. Countless e-mails land in the mailboxes of companies every day and many employees come into contact with them. Every new e-mail could be a potentially harmful e-mail. It is therefore important that they check their e-mails before they open unknown attachments or an attacker can access user data via a phishing page. In the event of suspicion, the users inform the IT help desk so that it can react accordingly and issue a company-wide

Table 1. Summary of Metrics for the Derivation of Habits

ID	Metric	Description
1	Latest password change	The timestamp of the latest password change.
2	Connection of usb devices	Logs whether external usb devices have been connected.
3	Data transfer to usb devices	Logs whether files have been transferred to external usb devices.
4	Access to suspicious domains	Logs if a suspicious or blacklisted URL has been accessed.
5	Screen Locking	The timestamps when the screen is locked.
6	Idle Time	The timestamps when the PC switches in idle state.
7	Information of Operating System	Information about Operating System changes, version, build number, etc.
8	Installed Software Versions	Logs Software versions and Software which is not whitelisted.

warning. In this way, the security awareness of the other employees can be increased and the correct reaction to the e-mail can be made. Only the frequent and regular implementation of the analysis of e-mails can be habitualized and can therefore also be measured with the SRHI. However, our metric regarding the access to suspicious domains can give a clue if an employee follows links in e-mails. Nevertheless, the analysis of e-mails cannot be totally measured by software.

Another behavior that is carried out quite often and is also subject to a certain regularity, is the screen locking. Users should always lock their screen when leaving their workplace. A look at everyday office life shows how often this is the case. Depending on their position in the company, employees may be out and about within the building more often, they may leave their workplace for a coffee or lunch break, to go to the toilet or for team meetings in seminar rooms. The most varied of constellations are conceivable - an unlocked screen offers the possibility of attack at any time. In Windows, the screen lock can be set quickly and easily using the Windows key + L shortcut. It is a simple key combination that can provide more security when leaving the workplace. As a rule, this is repeated regularly and frequently. A habit can definitely be established for this. The measurement for this behavior can be carried out with the SRHI. Moreover, the habit can also be measured automatically via software based on the metrics for screen locking and idle time. In contrast to the SRHI a software-based solution would also recognize idle times with unlocked screens.

At first glance, the use of removable storage devices is a behavior that cannot be measured in connection with habits. As already mentioned usb devices are a popular means of spreading malware, so their use should be avoided completely in the best case. Therefore, the desired behavior should rather be the *avoidance of usb devices*. Alternatively, certain places on network drives or the cloud can be used for the daily exchange of data between employees. If the required infrastructure is available, users can access it. Over a certain period of time, the data exchange can be habitualized via network drive or cloud. This behavior, on the other hand, can in turn be measured with the SRHI. The usb devices can be replaced in a certain way for data exchange. The security risk can thus be further minimized. Moreover, both metrics for connecting and transferring data to removable storage devices can be used to automatically measure this behavior.

In addition, updating the computer is another behavior that can be established in connection with a habit and can therefore be measured with the SRHI. Simply shutting down the computer after work and starting it the next morning can be such a habit. The computer installs updates automatically when it is shut down. Known vulnerabilities are thus usually closed. Shutting down the computer in the afternoon or evening can be established through a habit, which is then automatically triggered. A measurement of this behavior via a questionnaire and the SRHI is possible. Since a software can track the operating system version an automated measurement would be possible.

To sum up, we can easily derive the habits regarding screen locking, the avoidance of removable storage devices, as well as installing updates of the operating system. Even updates for other applications can easily be measured automatically. However, there exist many more security-related habits, which have to be analyzed regarding automation.

4.3. Habitualized Behavior Patterns

Since our software logs many behaviors of employees an analysis of the date can uncover habitualized behavioral patterns. Those patterns can then be used for targeted ISA trainings. On the one hand, an employee can have established habits that are compliant with the security policies. On the other hand, an employee can also be not compliant. Either way it is important for the employee to know and realize their habitualized behavior patterns.

Using artificial Intelligence (AI) was one idea for the analysis of the logged data. Since pattern recognition can be done with supervised or unsupervised learning we did not consider reinforcement learning approaches [28].

The first approach is to collect a large data set of productive data and let an AI automatically recognize patterns with unsupervised learning. The advantage of this approach is that the AI recognizes patterns that a human might overlook or cannot see at all. The second approach of supervised learning must be preceded by a manual data analysis. This sample data must be marked as a recognized pattern for training the AI.

However, since every employee has different behavioral patterns it is difficult to create patterns for supervised learning. Moreover, since unsupervised learning requires a large amount of data, the software would have to track the behavior of the employee for a long time period. Therefore, we decided against AI to uncover behavioral patterns and implemented a traditional algorithm instead.

The algorithm is intended to analyze the behavior metrics that are repeated on a daily basis. If a certain behavior occurs almost every day at a similar time, the algorithm should calculate the mean value of the behavior occurrence within the respective best time window. If a behavior occurs more frequently during the working days, the behavior could be considered as a habit.

The algorithm also checks behavior patterns which repeat every week. Behavioral patterns previously identified as daily habits should not be used to analyze weekly habits. If a certain behavior pattern occurs at a similar time of the week, the algorithm should calculate the respective best time window.

4.4. Software Design of the Tracking Tool

Since we have to implement a software solution which is compliant with the GDPR, we used a client server architecture. The client gathers the personal behavior information. The server only stores aggregated and anonymized data. Therefore, the server supports a dashboard which shows the aggregated numbers and metrics. However, it is not possible to receive information about a specific employee.

The client stores the personal behavior information only for the duration of behavior pattern detection. As soon as the patterns are recognized the detailed data is no longer required and will be deleted from the client. An employee can enforce the deletion of the gathered data at any time.

In future work the goal is to also train employees via the client. Therefore, the client should notify the employee about behavioral patterns so that the employee can actively try to avoid them. In order to allow the future improvement of our software we required a modular software which can easily be extended.

During our first iteration we only focused on Windows as an operating system. However, additional operating systems are also relevant. This is why we used python as a programming language. Furthermore, Python allows to easily use terminal commands. Since Python also supports the development of web applications, we could easily implement a frontend for the dashboard.

5. Implementation

This section describes the development of our artifact according to the methodology of Peffers et al. [18]. For the implementation of our software we used Python to easily roll out the prototype for different operating systems. However, in the first iteration we solely focused on Windows since it is often used in companies. In the following we give a short summary how each metric defined in Section 4.1 could be implemented under Windows using Python. Nevertheless, since we are using very context-specific solutions which could not be retrieved from literature, we are referencing web postings instead.

The last password change can easily be determined under Windows using the command `net user`. The command provides some information about the parameterized user name, including the date including the time of the last password change. Therefore, the current user name has to be determined. This functionality is already natively integrated in Python via the call `os.getlogin()`. In addition, the command can be expanded to also include the domain parameter. This is usually required if the current user is not a local Windows user and is instead logged on via a company domain [29]. The time of the last password change can be determined from the command line output. Finally, the date must be checked against the company's minimum time requirements. If the date of the last change is older than allowed, this incident will be logged.

The connection of removable usb devices can only be responded to directly under Linux with the help of udev-based libraries. No adequate solution can be found for Windows. Instead, Python can poll the WMI library to check which logical drives are connected to a computer and whether they are of the RemovableMedia type [30]. Only the drive letter, its size and the file

Figure 1. Dashboard for Assessment of Metrics

system used can be determined. For a better assignment, the name of the hardware is also interesting. In Windows this is called the `FriendlyName`. In order to be able to deduce the friendly name from a drive letter, several internal Windows databases must be queried. All `FriendlyNames` per connected device are noted in the table `Win32_DiskDrive`. In order to be able to infer the device used based on the drive letter, several tables must be linked with one another. On the one hand `Win32_DiskDriveToDiskPartition`, which links devices and partitions. On the other hand, `Win32_LogicalDiskToPartition`, which contains the drive letters of the partitions [31]. The combined data can be logged when a new removable medium is recognized.

Data transfers to removable usb devices are monitored. A common technique for this are so-called watchdogs on the file system. These trigger corresponding events for all file system operations, which can be conveniently handled. A watchdog functionality is built into most programming languages; this is also the case with Python. By recognizing the connection of a removable usb device, a separate watchdog can be generated for each device. Copying a file to a medium triggers the event `on_created`, which contains the file name as a parameter [32].

Tracking domain access of suspecious websites can be achieved by reading the local DNS cache of a Windows computer. To increase the performance of domain resolutions, Windows has its own internal DNS

cache. Domain names that have already been resolved are stored here with their assigned IP address in order to prevent the DNS from being queried in a short time. Using the command `ipconfig`, useful information can be read out via the network interface in Windows [33]. With the appropriate parameterization the command `displaydns` can read from the Windows-internal DNS cache.

The metrics screen locking, screen saver and idle time are related to each other: Both the activation of the screen saver and the locking of a computer can be easily tracked by low-level Windows APIs. Python can listen to internal system events of COM interfaces via `ctypes` [34]. The program interfaces used are unfortunately operating system-specific for Windows, but absolutely reliable. These events are then automatically logged. Exceeding the idle time, on the other hand, cannot be tracked via events. Instead, the library `pywin32` is used to periodically check how long ago the last user interaction with the computer was [35]. If the determined value exceeds the configurable limit value, this incident will be logged. The library `pywin32` can also be used to check whether the lock screen is currently active [36].

To track the version of the operating system, some commands and APIs for reading out the version number were viewed and compared [37]. Windows Management Instrumentation (WMI) turned out to be a positive result. WMI is one of the Win32 APIs and provides a command line interface for administration via CMD (WMIC) [38]. WMIC offers a query of different services with flexible parameterization. The command `wmic os get` is required to read the version information. All incidents will be logged if undesired operating system information or versions are detected.

The software versions and the installed software itself also represent critical metrics that need to be tracked. The WMIC is also used to determine those metrics. In this case, however, a different parameterization is required than for the operating system version. The specific command for reading out the software information is therefore: `wmic product get name, version` [39]. The resulting output shows the name of the installed software and its version numbers in a two-column table. This table can then be checked against a list of prohibited software and permitted minimum versions of installed software. Incidents are again logged.

In addition to the implementation of the logging mechanism and the gathering of metrics, we implemented a dashboard for the assessment of metrics. Figure 1 shows the dashboard. Filtering based on metric category or time ar possible. The same dashboard can be used to configure the priorities as well as allowed software versions.

6. Results Discussion

We based our research on the design science research methodology proposed by Peffers et al. [18]. We started by identifying a problem which lies in the nature of human behavior: it is difficult to quantify. Then, we gathered the requirements for the measurement of habits as one factor of the ISA of employees. The most important requirement for the experts was the automation. Therefore, we defined our objectives regarding the automated measuring of habits.

In Section 4 and Section 5 we summarized our design and development phase. The resulting artifact can now be used to answer our research questions.

RQ1 focuses on the measuring of habits in the context of ISA. Since the SRHI is the de facto standard questionnaire to measure habits, it is possible to create such questionnaires with security-compliant behaviors. However, the answers of the employees could differ from their actual behavior. This is why software based behavior tracking is a proposing alternative. Consequently, we defined metrics which can be used to track some security related behaviors. The measurement results can be used to get an idea about established habits of the user. However, it is difficult to decide whether a behavior is a habit solely by tracking it via software. Therefore, we recommend to combine the software tracking with interviews or questionnaires to determine if the actual behavior represents a habit.

RQ2 focuses on the automated measurements. Since we use a software to track the behavior of employees, it is possible to fully automate the retrieval of metrics. However, it is not possible to automatically decide whether a certain behavior represents a habit. This is due to the complexity of habits. The combination of metrics can lead to assumptions about habits, but one metric alone cannot be used to decide whether an employee has a habit. For example, the metric for screen locking cannot be used isolated to decide if the employee has the habit to lock their screen. Additional information is required: maybe the employee has turned on the auto-lock after a certain idle time. This is why the metric for idle times has to be considered as well. If on the one hand no incident regarding the idle time is recorded, but the screens have been locked, we could assume the employee has the habit to lock their screen.

RQ3 focuses on privacy aspects of our software artifact. We implemented several functionalities to ensure privacy: We aggregated the metrics on the server

without any possibility to get information about the actual employee causing the numbers. The client keeps the data as short as possible so that an attack on an employee's computer cannot reveal all of the tracked behavior. Of course some additional permissions will be required in different companies. Nevertheless, we argue that an GDPR-compliant automation is possible.

After the development of our artifact, we reached out to companies to start the demonstration process of Peffers et al. [18]. Overall the companies liked the idea and wanted to conduct a piloting project. We will use the pilots for an extended evaluation of our artifact. However, we already received some feedback, which we will use as a possible research entry point for our future work.

Our metrics cannot be used for every employee identically: Some employees do not have administrator privileges on their computers which renders the metrics for *operating system version* and *software versions* useless. Those metrics can only be useful in companies which do not have centralized patch managements or for employees with administrator privileges. Moreover, some companies may use watchguards or strong firewalls which will block the access to suspicious domains. However, even if the access was blocked it could be interesting to have a metric representing the access attempts.

7. Conclusion

In this paper, we summarized our process iteration for the development of an artifact for automated measurement of habits in the context of ISA. Therefore, we used the design science research methodology proposed by Peffers et al. [18]. In the concept we presented our design for the software which consists of the definition of metrics, the derivation of security-relevant habits as well as the recognition of additional behavioral patterns. Moreover, we summarized the architecture of our software tool. Afterwards, we explained how every metric can be measured under Windows.

Since our prototype has only been evaluated by the presentation to experts of companies, we will carry out a large-scale evaluation in the next steps of our research. Therefore, we will install the prototype within different companies and will track the behavior of employees. In parallel, the companies will sensitize their employees for ISA. In an ideal world, the metrics will then improve due to the ISA trainings. During and after the evaluation project, we will gather feedback to improve the prototype within the next process iteration.

During the evaluation, we will also continue the development of the software to enable the use on additional operating systems. Moreover, we will implement additional features for the sensitization of employees based on their locally tracked behavior.

Acknowledgements

Authors ... were supported by the ...

References

[1] R. Heartfield and G. Loukas, "Detecting Semantic Social Engineering Attacks With the Weakest Link: Implementation and Empirical Evaluation of a Human-as-a-security-sensor Framework," *Computers & Security*, vol. 76, pp. 101–127, 2018.

[2] A. E. Schütz, "Information security awareness: It's time to change minds!," in *Proceedings of International Conference on Applied Informatics Imagination, Creativity, Design, Development - ICDD 2018*, 2018.

[3] H. C. Triandis, *Interpersonal behavior*. Monterey Calif.: Brooks/Cole, 1977.

[4] E. Parliament, "General Data Protection Regulation (GDPR) – Official Legal Text." https://gdpr-info.eu/, 2016. Last accessed: 2021-06-15.

[5] M. Bada and A. Sasse, "Cyber Security Awareness Campaigns: Why do they fail to change behaviour?," July 2014. Num Pages: 38 Place: Oxford, UK Publisher: Global Cyber Security Capacity Centre, University of Oxford.

[6] D. E. Montaño and D. Kasprzyk, "Theory of Reasoned Action, Theory of Planned Behavior, and the Integrated Behavioral Model," in *Health Behavior and Health Education. Theory, Research and Practice. 4th Edition* (K. Glanz, B. K. Rimer, and K. Viswanath, eds.), pp. 67–96, San Francisco, CA: Jossey-Bass, 2008.

[7] L. Hirshfield, P. Bobko, A. J. Barelka, M. R. Costa, G. J. Funke, V. F. Mancuso, V. Finomore, and B. A. Knott, "The Role of Human Operators' Suspicion in the Detection of Cyber Attacks:," *International Journal of Cyber Warfare and Terrorism*, vol. 5, pp. 28–44, July 2015.

[8] B. Verplanken and H. Aarts, "Habit, Attitude, and Planned Behaviour: Is Habit an Empty Construct or an Interesting Case of Goal-directed Automaticity?," *European Review of Social Psychology*, vol. 10, no. 1, pp. 101–134, 1999. Number: 1.

[9] B. Verplanken, "Beyond frequency: Habit as mental construct," *British Journal of Social Psychology*, vol. 45, pp. 639–656, Sept. 2006.

[10] H. Aarts and A. Dijksterhuis, "Habits as knowledge structures: Automaticity in goal-directed behavior," *Journal of Personality and Social Psychology*, vol. 78, no. 1, pp. 53–63, 2000. Number: 1.

[11] W. Wood, L. Tam, and M. G. Witt, "Changing Circumstances, Disrupting Habits," *Journal of Personality and Social Psychology*, vol. 88, no. 6, pp. 918–933, 2005. Number: 6.

[12] M. F. Ji and W. Wood, "Purchase and Consumption Habits: Not Necessarily What You Intend," *Journal of Consumer Psychology*, vol. 17, no. 4, pp. 261–276, 2007. Number: 4.

[13] B. Gardner, "A review and analysis of the use of 'habit' in understanding, predicting and influencing health-related behaviour," *Health Psychology Review*, vol. 9, pp. 277–295, Aug. 2015.

[14] B. Verplanken and S. Orbell, "Reflections on Past Behavior: A Self-Report Index of Habit Strength," *Journal of Applied Social Psychology*, no. 33, pp. 1313–1330, 2003.

[15] R. J. H. van Bree, M. M. van Stralen, C. Bolman, A. N. Mudde, H. de Vries, and L. Lechner, "Habit as moderator of the intention-physical activity relationship in older adults: a longitudinal study," *Psychology & Health*, vol. 28, no. 5, pp. 514–532, 2013.

[16] T. Fertig and A. Schütz, "About the Measuring of Information Security Awareness: A Systematic Literature Review," in *53rd Hawaii International Conference on System Sciences*, Jan. 2020.

[17] A. L. Edwards, *The social desirability variable in personality assessment and research*. The social desirability variable in personality assessment and research, Ft Worth, TX, US: Dryden Press, 1957.

[18] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007. Publisher: Taylor & Francis.

[19] T. Fertig, A. E. Schütz, and K. Weber, "Current Issues Of Metrics For Information Security Awareness," in *Proceedings of the 28th European Conference on Information Systems (ECIS), An Online AIS Conference*, 2020.

[20] M. Wilson and J. Hash, "Sp 800-50. building an information technology security awareness and training program," tech. rep., National Institute of Standards & Technology, Gaithersburg, MD, USA, 2003.

[21] V. Poitevin, "Should companies ban USB devices?." https://www.stormshield.com/news/a-future-without-usb-sticks, 2019. Last accessed: 2021-06-15.

[22] M. Tischer, Z. Durumeric, S. Foster, S. Duan, A. Mori, E. Bursztein, and M. Bailey, "Users really do plug in USB drives they find," in *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016*, pp. 306–319, Institute of Electrical and Electronics Engineers Inc., 2016.

[23] L. Zinatullin, *The Psychology of Information Security: Resolving conflicts between security compliance and human behaviour*. IT Governance, 1. edition ed., 2016.

[24] K. Dilanian, "Officials: Edward snowden took NSA secrets on thumb drive." https://www.latimes.com/politics/la-xpm-2013-jun-13-la-pn-snowden-nsa-secrets-thumb-drive-20130613-story.html, 2013. Last accessed: 2021-06-15.

[25] K. C. Laudon and C. G. Traver, *eCommerce 2016: Business, Technology, Society*. Pearson Education, 12th edition ed., 2016.

[26] StatCounter, "Desktop operating system market share worldwide." https://gs.statcounter.com/os-market-share/desktop/worldwide, 2020. Last accessed: 2021-06-15.

[27] A.-A. Hariri and M. Powell, "Zero day initiative — CVE-2020-9715: Exploiting a use-after-free in adobe reader." https://www.thezdi.com/blog/2020/9/2/cve-2020-9715-exploiting-a-use-after-free-in-adobe-reader, 2020. Last accessed: 2021-06-15.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[29] Microsoft, "Net user." https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc771865(v=ws.11), 2016. Last accessed: 2021-06-15.

[30] S. White, D. Batchelor, D. Coulter, cofeeyen, M. Jacobs, and M. Satran, "Win32_logicaldisk class - win32 apps." https://docs.microsoft.com/en-us/windows/win32/cimwin32prov/win32-logicaldisk, 2018. Last accessed: 2021-06-15.

[31] K. Adrianus, "Python: get name of a USB flash drive device [windows]." https://stackoverflow.com/questions/33784537/python-get-name-of-a-usb-flash-drive-device-windows, 2019. Last accessed: 2021-06-15.

[32] D. Mastromatteo, "How to create a watchdog in python to look for filesystem changes." http://thepythoncorner.com/dev/how-to-create-a-watchdog-in-python-to-look-for-filesystem-changes/, 2019. Last accessed: 2021-06-15.

[33] J. Mueller, *Windows Command Line Administration Instant Reference*. Serious skills, Wiley, 2010.

[34] R. Upole, "[python-win32] ISensLogon COM object." https://mail.python.org/pipermail/python-win32/2008-January/006645.html, 2008. Last accessed: 2021-06-15.

[35] Derek, "winapi - detecting idle time using python." https://stackoverflow.com/questions/911856/detecting-idle-time-using-python/33765403#33765403, 2015. Last accessed: 2021-06-15.

[36] Fretxik, "In python 3, how can i tell if windows is locked?." https://stackoverflow.com/a/57955010/3123870. Last accessed: 2021-06-15.

[37] R. Srinivasan, "How to find your windows 10 build number, version, edition and bitness." https://www.winhelponline.com/blog/find-windows-10-build-version-edition-bit/, 2016. Last accessed: 2021-06-15.

[38] S. White, D. Batchelor, D. Coulter, A. Patridge, and M. Satran, "wmic - win32 apps." https://docs.microsoft.com/en-us/windows/win32/wmisdk/wmic, 2018. Last accessed: 2021-06-15.

[39] S. Steiger, "windows - WMI: Get the list of installed softwares." https://stackoverflow.com/questions/25131413/wmi-get-the-list-of-installed-softwares/25132527#25132527, 2014. Last accessed: 2021-06-15.