TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Darya Harachka 184053IVSB

Securing a Semi-Attended Self-Service Kiosk based on the Example of the Services offered by Apollo Digital

Bachelor's thesis

Supervisor: Priidu Paomets

Master of Science

Tallinn 2021

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Darya Harachka 184053IVSB

Iseteeninduskioski lahenduse turvalisuse analüüs Apollo Digitali näitel

Bakalaureusetöö

Juhendaja: Priidu Paomets

Tehnikateaduste magister

Tallinn 2021

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Darya Harachka

17.05.2021

Abstract

Unattended and semi-attended self-service kiosks have become a popular retail solution in the modern world due to simplicity of use and ability to provide contactless service. Even though there are standards that are covering the security regulations for payment and health kiosks, there are not many sources that would address the security of self-service kiosks in retail as a complex.

This thesis paper covers the analysis of the physical and software components of the self-service kiosk setup used in Apollo Digital, risk assessment conducted using the list of assets identified for both current and new generations of the kiosk software design, and provides suggestions for controls to be implemented for the unacceptable risks. The critical vulnerabilities identified as the result of risk assessment are mostly arising from misallocation of both system and kiosk application privileges, and from the mechanism for providing privileged system access.

The set of controls proposed to address the risks identified covers the design of the system of access rights for various groups of kiosk setup users, the development of the authenticator application utilizing the time-based one-time password mechanism, measures to secure the new generation of software utilizing Electron JS framework, and adjustments to be made to the business process to eliminate the need in the vulnerable software components. The implementation of the controls described should provide the treatment to all the unacceptable risks identified, and enhance the security of the semi-attended self-service kiosk used in the Apollo Digital, and could be used as a part of the security model for the similar setups.

The research covers the security aspects of the self-service kiosk machine itself and its components, excluding connection mechanisms and the external services it is connected to.

This thesis is written in English and is 62 pages long, including 5 chapters, 5 figures and 3 tables.

Annotatsioon
Iseteeninduskioski lahenduse turvalisuse analüüs Apollo Digitali näitel

Pool- või täiesti järelvalveta iseteeninduskioskid on muutunud tänases jaekaubanduses väga populaarseks tänu oma kasutusmugavusele, -lihtsusele ja võimalusele teha oste kontaktivabalt. Kuigi eksisteerib erinevaid maksete ja tervishoiu valdkonna regulatsioone ning standardeid, siis pole kuigi palju selliseid allikaid, mis adresseeriks iseteeninduskioskide turvalisuse aspekte terviklikult.

Käesolev lõputöö käsitleb Apollo Digitali poolt kasutatavate iseteeninduskioskide riist- ja tarkvaralisi komponente, teeb riskianalüüsi nii hetkel kasutuses oleva, kui uue loodava generatsiooni kioski tarkvara kohta ning toob välja soovitused mitteaktsepteeritavate riskide maandamiseks ja puudujääkide eemaldamiseks. Tuvastatud kriitilised haavatavused on tingitud peamiselt operatsioonisüsteemi ja rakenduse õiguste ebapiisavast haldusest ja süsteemile priviligeeritud ligipääsu võimaldamisest.

Töö tulemusena välja pakutud lahendused sisaldavad süsteemi ligipääsu õiguste määramist erinevatele kasutaja gruppidele, ajapõhiseid ühekordseid paroole kasutava autentimisrakenduse arendamist, loodava Electron.js-põhise rakenduse turvamise soovitusi ning ettepanekuid äriprotsessidesse sisse viidavate muudatuste kohta, mis elimineeriks haavatavate komponentide kasutusvajaduse. Toodud soovituste rakendamine peaks pakkuma lahenduse kõikidele tuvastatud mitteaktsepteeritavatele riskidele ning tõstma Apollo Digitali poolt loodavate pool- või täiesti järelvalveta isenteeninduskioskide turvalisust. Seda analüüsi saab kasutada ka teiste sarnaste lahenduste juures.

Uurimustöö katab iseteeninduskioski enda ning selle komponentide turvalisuse aspekte, kuid ei vaatle väliseid teenuseid ega ühendusi nendega.

Lõputöö on kirjutatud Inglise keeles ning sisaldab teksti 62 leheküljel, 5 peatükki, 5 joonist, 3 tabelit.

List of abbreviations and terms

PCI	Payment Card Industry
IT	Information Technology
MDF	Medium-Density Fibreboard
NFC	Near Field Communication (technology)
QR code	Quick Response code
LAN	Local Area Network
PTS	PIN Transaction Security
PCI SSC	PCI Security Standards Council
POI	Point of Interaction
UAC	User Access Control is a mandatory access control enforcement feature implemented in Windows operating system, the purpose of which is to improve the security of Microsoft Windows by limiting application software to standard user privileges until an administrator authorizes an increase or elevation.
Microsoft Defender SmartScreen	Microsoft Defender SmartScreen is a security feature allowing to protect against phishing or malware websites and applications, and downloading of potentially malicious files.
UWP	Universal Windows Platform is a computing platform created by Microsoft and designed to help develop universal applications that perform uniformly on all Windows devices.
.NET framework	The .NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows.
HTTPS	Hypertext Transfer Protocol Secure
TLS	Transport Layer Security
POS	Point of Sale in the context of this paper refers to the business location at which a self-service kiosk is installed.
UI	User Interface
API	Application Programming Interface
WCF	Windows Communication Foundation is an open-source runtime and a set of APIs in the .NET Framework for building connected, service-oriented applications.
TCP	Transmission Control Protocol is a highly reliable host-to-host protocol between hosts in packet-switched computer

	communication networks, and in interconnected systems of such networks.
HTTP	Hypertext Transfer Protocol
Azure AD	Azure Active Directory is an enterprise cloud-based identity and access management solution developed by Microsoft.
React	React is an open-source, front end, JavaScript library for building user interfaces or UI components.
Electron	Electron is an open-source software framework developed and maintained by GitHub, that allows for the development of desktop graphical UI applications using web technologies.
APM	Application Performance Management
REST	Representational State Transfer is an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other.
gRPC	gRPC is a language agnostic, high-performance Remote Procedure Call (RPC) framework.
GraphQL	GraphQL is an open-source data query and manipulation language for APIs, and a runtime for fulfilling queries with existing data.
SQLite	SQLite is an open-source, zero-configuration, self-contained, stand-alone, transaction relational database engine designed to be embedded into an application.
SDK	Software Development Kit
ISO/IEC 27001 standard	ISO/IEC (International Organization for Standardization/ International Electrotechnical Commission) 27001 is an international standard on how to manage information security.
Node.js	Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that allows to execute JavaScript code outside a web browser.
TOTP	Time-based One-Time Password
Google Authenticator	Google Authenticator is a software-based authenticator designed by Google that implements two-step verification services using the Time-based One-time Password Algorithm and HMAC-based (Hash-based Message Authentication Code) One-time Password algorithm, for authenticating users of software applications.
Nano ID	Nano ID is a secure, URL-friendly, unique string identifier generator producing alphanumeric strings of 108 bits as an output.
GUID	Globally Unique Identifier is a 128-bit number used to identify information in computer systems.

NIST	National Institute of Standards and Technology is a physical sciences laboratory and a non-regulatory agency of the United States Department of Commerce.
CSPRNG	Cryptographically Secure Pseudorandom Number Generator
SHA-3	Secure Hash Algorithm version 3
Security strength	Security strength is a number associated with the amount of work (i.e., the number of operations) that is required to break a cryptographic algorithm or system.
Second preimage	A message X', that is different from a given message X , such that its message digest is the same as the known message digest of X.
Second preimage security	Second preimage security or resistance an expected property of a cryptographic hash function whereby it is computationally infeasible to find a second preimage of a known message digest.
Collision	An event in which two different messages have the same message digest.
Collision security	Collision security or resistance expected property of a cryptographic hash function whereby it is computationally infeasible to find a collision.

Table of contents

List of figures

List of tables

1 Introduction

The spread of self-service kiosks caused the necessity for developing new security features for existing services. As the setups in question are developed to function in unattended or semi-attended environments, service providers need to ensure that end clients can use services provided only in an intended way. It also raised the question of providing physical security and security of the underlying operating system for all the elements of self-service kiosks, as opposed to using services through attended service points or online from users' personal devices. Therefore, it became necessary to enhance the existing approach to securing services provided to accommodate for the specifics of the semi-attended self-service setup.

This thesis paper aims to identify the risks arising from the operation of semi-attended self-service kiosks developed by the Apollo Digital and used in many companies, both inside of the Apollo Group, and outside, assess the probability of them occurring and the severity of the effect they may cause, and propose potential mitigation mechanisms to address the risks that are unacceptable for the organization. The scope of the paper covers the self-service machine itself with its physical and software components, excluding the external services the setup uses and the security of the connection to those services.

The relevance of the topic in question is defined by the fact that the introduction of semi-attended self-service kiosks is an ongoing project for an organization, and there is a need to study potential threats to the setup and ways to mitigate them. Apollo Digital provides self-service kiosk setups that are used for various businesses in a set of countries for providing retail, entertainment and food industry related services. The countries currently using the self-service kiosk solution designed by the organization include Estonia, Latvia, Lithuania, and Finland. The kiosks are usually placed on the premises belonging to the businesses owning them to provide additional service points for the end customers. The operations that can be performed by clients using the setup include registration with the user account for businesses supporting such functionality, selection of items to be purchased from the kiosk menu or using pre-order proof, and payment for the items or services selected.

The self-service kiosk industry has been increasingly developing over the last 30 years within various spheres of business including health industry, hospitality, payment systems, and others. However, there is no general standard that would be specific to the security of kiosk solutions. The current set of standards and recommendations that are related to kiosk security includes the following:

- ADA (Americans with Disabilities Act) Standards for Accessible Design [1] defining the requirements for kiosk accessibility for people with disabilities
- HIPAA (Health Insurance Portability and Accountability Act) Standards for Privacy & Self-Service [2] designed for providing privacy of the data in the healthcare organizations
- PCI (Payment Card Industry) [3] & EMV (Europay, Mastercard, and Visa) Payment Standards designed for regulating the security of payment systems
- UL (Underwriters Laboratories) Standards defining safety requirements for the physical setup

There are also local standards and guidelines related to the kiosk security that are defined within organizations for specific self-service kiosk setups, such as The University of Texas Rio Grande Valley Kiosk Security Standard [4], or Berkeley University of California Campus Guidelines for Kiosk Workstations [5]. Additionally, the security of various physical and software components included with the self-service kiosk setup is covered separately by more general security standards and recommendations, as these components are commonly represented in other IT (Information Technology) systems.

Even though there are no general security guidelines that would be accumulated in a single exhaustive document for the security within the self-service kiosk industry, there exists a multitude of commercial solutions designed to enhance the security of kiosk installations. The most popular complex kiosk software security solutions include SiteKiosk software [6], KioWare software [7], and similar. These tools support restricting underlying system capabilities and tools that have a potential of exposing additional functionality to unauthorized users, user management, keyboard filtering, custom user interface management, and remote monitoring and control over the kiosk system, providing a comfortable complex solution to most common security issues arising from the operation of self-service kiosks. Another popular solution to secure the access to the

underlying system involves using special Windows OS kiosk mode option [8] restricting the access to a single kiosk application.

However, the solutions described above are not specific to particular kiosk installations, do not support all types of kiosk applications, and come with either license or subscription fee. Due to incompatibility of the kiosk application currently used within the organization with most technologically and financially suitable solutions available on the market, the problem of securing the self-service kiosk used by Apollo Digital is remaining actual.

The fact that there is no constant control over the operation of the kiosk setups leaves a potential for individuals interacting with them to perform unauthorized actions. With that many businesses using the service in question it is crucial to ensure proper security level in order to prevent data loss and theft, and any unintended exposure of the system as it may result in affecting multiple companies along with their client base at once.

The structure of this thesis consists of the following four chapters:

- **Description of the self-service kiosk setup chapter** describes the variety of setups used in the organization, along with the set of physical and software components and security measures implemented
- **Risk assessment chapter** presents the analysis of the set of assets associated with the self-service kiosk installation, identifies the threats and vulnerabilities related to those assets, provides evaluation for the risks that do not have controls in place, and defines the unacceptable level of risk for the setup
- **Design of risk treatment proposal chapter** suggests controls that can be implemented in order to address the unacceptable risks identified in the previous chapter
- **Summary chapter** describes the main goal of the thesis and the most significant outcomes of the research conducted

2 Description of the semi-attended self-service kiosk setup used in the organization

The self-service kiosk setups provided by Apollo Digital are designed to be installed in various semi-attended locations, such as cinemas, bookstores, shopping mall grounds, cafes and restaurants. Independent of the exact installation area, the placement of the kiosks is covered by video surveillance. For the setups installed on the premises of the companies using the service, the kiosks are usually located within a couple of meters from human-operated counters in order to ensure quick response to end customers' requests or any problems connected to the kiosk operation. Physical security and surveillance services are handled by the security staff of the building, or a particular company in case of cinemas and restaurants.

The main purpose of the setup is to provide end customers with an ability to select and purchase goods and services provided by a company, covering ticket sales, various course orders for restaurants and cafes, and product sales for the bookstores. The setups located on the territory of shopping malls are also used for sales by the services described above.

The system used is constantly evolving with newer components becoming available and technology advancing, therefore in this paper there will be described both the existing setup and the adjustments to it that has already been approved and scheduled for implementation.

2.1 Description of physical kiosk components

The form of the self-service kiosk station may vary depending on the country and actual installation place; however, it provides access to the same generalized functionality. The physical differences between the setups used include the screen orientation, the form and material of the enclosure box, and the layout of the hardware parts accessible to the end user. Sample version of the setup used in bookstores in Estonia where the hardware is installed into a specially designed MDF (Medium-Density Fibreboard) box with the front

part covered by a metal plate and with barcode reader and cash recycling system mounted into the furniture next to the kiosk box is represented at Figure 1.

Figure 1. Example: Self-service kiosk setup used in bookstores in Estonia.

An example of the setup used in cinemas in Finland using a custom-made metal case and built-in barcode scanner device with no cash recycling system incorporated is represented at the Figure 2.

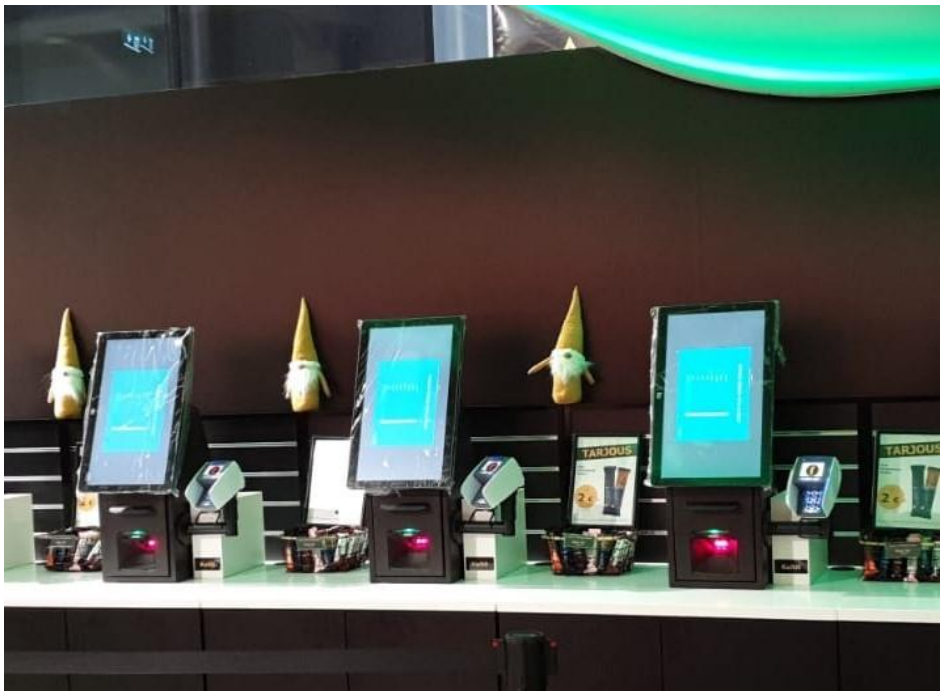

Figure 2. Example: Self-service kiosk setup used in cinemas in Finland.

The self-service kiosks used in the countries listed above share a common set of the following hardware and physical components:

- Case for hardware installation
- Touchscreen computer
- Cash recycling system for the setups installed in Latvia, Estonia and Lithuania
- Receipt printer
- Barcode reader
- Unattended payment terminal system
- Fiscal memory module for the kiosk setups installed in Latvia and Lithuania
- Connecting cables dependent on the actual setup

There are two custom-made options used for the hardware installation cases: one is made of MDF with the front side enforced by a metal plate that is used for installations in Estonia, Latvia, and Lithuania, and the second one fully made of metal used in Finland. In all setups the case is screwed to the furniture item it is placed upon. Access to the internals of the case box is secured by a mechanical lock, and the keys for it are stored with the shift administrator of the establishment. These keys are used by the POS (Point of Sale) staff to get access to cash compartments for cash collection or loading, or for replacing the printer paper rolls. An additional set of keys is also provided to the technical service company for maintenance purposes.

The touchscreen computers used in all the countries are provided by Elo Touch Solutions and belong to their 22- or 15-inch l-series for Windows providing an ability to set the screen on a vertical or horizontal mount for countertop setups. While the touchscreen part of the system is exposed to the end clients, all the computer ports are hidden and sealed by a plastic panel in order to restrict unauthorized access to them.

The compact cash recycling system used for kiosk setups in Lithuania, Latvia, and Estonia consists of coin and banknote recyclers with attached cash storage and can also come in two model options - CI-10 and CI-5 produced by Glory Global Solutions. Both models come with inbuilt counterfeit detection using ultraviolet and magnetic sensors. The system is configured for accepting and dispensing euro bills and coins, as at the moment the kiosks are used only in the countries which have euro as the official currency.

Connection between the cash recycling system and the kiosk system itself is managed via Ethernet cable placed inside the hardware case box.

Physical access to the internals of the cash recycling system cash compartment is restricted by a mechanical lock that can be accessed only from the internals of the hardware case for the current setup. There is an additional feature within the self-service application designed to allow the compartment to be first unlocked through the software, and then physically by using keys. There is a set of three different keys used – one for pulling the cash system out of its enclosing box, another one for getting access to the removable cash box, and the third one used to open the removable cash box. The keys are stored with the shift administrator of the establishment in order for employees to be able to collect cash or refill the machine.

The kiosk uses VKP80II SX receipt printer produced by Custom S.P.A. that is connected to the rest of the system via USB, or serial cable in case it is connected to the fiscal device. The printer is also connected to the network via an Ethernet cable that allows to use the printer's internal web server for its remote management. The device itself is bolted inside the hardware case which has a slit for issuing the receipts to clients.

The kiosk system also incorporates a barcode reader which depending on the setup can be installed either within the actual kiosk case or mounted onto the table next to the box. The scanner model used is Magellan 3300HSi provided by Datalogic. It is connected to the system via a USB cable, that is placed securely either within the box case or locked table counter. The device allows the kiosk system to read and recognize barcodes and QR (Quick Response) codes presented in both printed and digital formats (used for code recognition from mobile devices screens).

The unattended payment terminal system used for kiosk setups consists of separate devices connected via Ethernet cables into a cabled LAN (Local Area Network). It has three main components: an NFC (Near Field Communication) reader, a chip and magnetic stripe card reader for processing bank and ID cards, and a pin pad. These devices are bolted into the metal panel in the front of the kiosk hardware case, and the installation requires validation by a specialist from an authorized technology provider, after which the system can be activated.

All three payment terminal system components contain anti-tamper mechanisms that will be triggered in case a physical penetration attempt of the device is detected. These mechanisms are based on using a set of pressure detectors that are configured to react to shaking or pressure change in case the device gets removed from the setup. The activation of tamper mechanisms puts the device into out of service state, which implies that the device is assumed to be tampered with, stops its operation, and requires reactivation after maintenance and security checks [9].

All the components of the unattended payment terminal system are provided as a single system of Ingenico iSelf PCI-PTS (Payment Card Industry PIN Transaction Security) 4.x certified series. PCI PTS POI (Point of Interaction) standard is a part of the series of standards designed by PCI SSC (PCI Security Standards Council), a global forum that brings together payments industry stakeholders to develop and drive adoption of data security standards and resources for safe payments worldwide [3]. This standard defines a set of physical and logical POI device security requirements that need to be fulfilled for the device to become certified.

Fiscal memory modules are required to be incorporated into self-service kiosk setups installed on the territory of Lithuania according to the Order on Rules for the Use of Cash Machines and Direct Communication Computer Network Terminals and Approval of the Form of the Decision to Register a Cash Device [10], and for Latvia according to Regulations Regarding Technical Requirements for Electronic Devices and Equipment for the Registration of Taxes and Other Payments [11]. The main purpose of those devices is to store records for all the transactions made via the kiosk system, which can later be exported and analysed by the government in order to record sales tax owed to the country. The connection between the fiscal device and the kiosk system is implemented via a USB cable. The devices themselves use RS-232 serial interface, but since many modern computers do not come with such ports available, the setup includes a virtual USB-to-COM adapter. Every transaction received from the kiosk system used in Latvia and Lithuania that requires issuing a receipt is registered the fiscal module where its details get recorded. The data stored with the fiscal module can also be exported via network connection to the external storage to be stored as a backup. These operations along with maintenance of the fiscal device are fully outsourced to the IT and technical maintenance service.

The kiosk system components are powered by a set of power cables protected by the piece of furniture on top of which the kiosk is mounted and the hardware case itself. Separate power cables are used for powering the touchscreen computer, receipt printer, payment terminal system and cash recycling system, which uses three power cables for its components. The network connection is managed via a set of three Ethernet cables connected to the touchscreen computer, payment terminal system and cash system.

2.2 Description of software components and system operation

On the software side there are also several common components that are currently used for all the kiosk installations:

- Windows 10 operating system
- Self-service kiosk application
- Hardware services
- Print server application
- Remote system access application
- System monitoring software

As far as the kiosk system is placed in a semi-attended environment where there is no constant monitoring implemented over its operation, it is crucial that the functions available to end clients are limited to the intended minimum. Therefore, there is a set of limitations set on the functionality of Windows operating system used. The exact implementation varies based on the country of installation due to restrictions existing in local regulation. The minimum version of Windows 10 used for all the countries is build 16299 (version 1703), also known as Fall Creators Update. The choice of Windows OS to be used with the setup was dictated by the technological stack used for developing the kiosk application, which is specific to Windows platform.

For Latvia and Lithuania, a stripped and locked out version of Windows is used, where the system boots directly into the self-service kiosk application and Windows functions are not available. The rest of the services mentioned above are running in the background mode.

In the Estonian setup there is a regular version of Windows 10 used with some restrictions put via disabling undesired Windows capabilities, including the following:

- Microsoft Consumer Experiences service which is providing recommendations and suggestions for additional application installation
- Edge swiping behaviour for switching between the applications
- Windows UAC (User Access Control) feature
- Microsoft Defender SmartScreen feature
- Windows Automatic Updates feature

The self-service kiosk application is a client facing UWP (Universal Windows Platform) application designed based on .NET framework with the main purpose to provide end users with the access to the main kiosk functionality. Depending on the business using the kiosk it can expose some set or all of the following operations available to users:

- Product or course purchase
- Ticket purchase
- Reservation redeeming
- Invoice payment for the food orders issued from the cinema
- Previously purchased ticket printing

The UWP platform runs applications in an AppContainer security context. AppContainers are sandboxes that isolate the runtime environment for a process or any code that runs in that process. Their goal is to block any possibility for malicious code to get inside the process address space [12]. Additionally, UWP applications have less access rights than "Standard User" and can never run with administrator privileges. Such applications can request additional capabilities to get a few more rights with permission from the user but have limited access to the system, user data and devices. While this approach provides higher level of security out of the box it also introduces limitations on using UWP applications in combination with other software products due to their sandboxed execution. To enable the operating system to run the UWP self-service application the Windows Developer Mode is enabled on the kiosk machines.

The connection between the self-service application and the remote API is managed over HTTPS (Hypertext Transfer Protocol Secure) using TLS (Transport Layer Security) protocol version 1.2.

User input to the application in cases users are required to provide barcode or voucher numbers is managed via an on-screen numeric keyboard with a limited set of symbols

available that is built into the application. The inbuilt Windows keyboard is however used in Estonian and Finnish setups to allow users to input the promotion code for their purchase.

Hardware services is a service application containing plugins for hardware interfaces responsible for the interaction with payment terminal system, fiscal device, ID card reader support, and POS printing at low level. The UI (User Interface) of the application is implemented via WinForms, and it exposes its functionality to the rest of the system through WCF (Windows Communication Foundation) TCP (Transmission Control Protocol) connections to ensure reliable communication over an established session.

The printing capabilities of the kiosk are managed separately by the print server application, a WinForms application that is using WCF over HTTP (Hypertext Transfer Protocol) for communication. The main purpose of this application is to expose print commands to the self-service application, retrieve data from the database and issue print commands to the printer device. The print server application was implemented due to the fact that UWP applications do not have access to printing capabilities by default and requires user confirmation on every print command, which would expose access to the operating system functionality to end users.

When the self-service kiosk machine boots up, it executes a script which starts with executing hardware services and print server programs before proceeding to the execution of the self-service kiosk application itself in order to ensure that all the hardware is available. In case any services that the self-service application is required to communicate with are detected as not available the status check page is displayed with a set of “Retry” and “Ignore” options allowing either to try to identify the required services once more, or to proceed with execution. If there are any problems identified with the hardware during the operation of the kiosk, then the self-service application enters the “Out of order” state and human interaction is required to check the issue, fix it, and to restart the machine and the service, if needed.

The application used for enabling remote access is TeamViewer. It is started on the system boot and running in the background and is used by the technical maintenance service provider in order to get remote access to the kiosk system.

Logging for the setup is managed at two different levels. For hardware services and print server logs are kept on the local machine in the text format. Print server activity along with self-service application logging is also processed by Azure Application Insights cloud service, an extensible APM (Application Performance Management) service designed to monitor live applications, automatically detect performance anomalies, and use an inbuilt set of powerful analytics tools to help diagnose issues and analyse user activity. The level of logging for hardware services can be adjusted according to the needs. At the most detailed level, each call to a device API is logged, along with all the parameters and responses. The most sensitive data being logged is contained in the logs for fiscal and card payment calls, including full receipt info, and full details of card payment results. However, the payment terminal system is designed to mask card numbers and other sensitive fields in order to ensure PCI compliance. The access to the device logs is shared only to the IT and technical maintenance organizations.

The monitoring of the system for Finnish and Estonian setups is managed via Miradore asset management system software that is integrated with Azure AD (Azure Active Directory) service. It issues status alerts including the information about the current user of the system, its status, software set and patching level that are in order reviewed by the IT and technical maintenance service provider specialists. In Finland there is an FSecure PSB (Protection Service for Business) additionally installed that allows to monitor the devices, apply custom security profiles to selected devices, view reports and statistics regarding threats prevented by the software, and download and distribute software updates [13]. For the installations in Latvia and Lithuania the monitoring is fully outsourced to the maintenance provider. However, at the moment the monitoring solutions are tracking only the state of the system as a whole, but not the status of the self-service application itself.

In order to identify software or hardware failures there is a scripted job running in the background that is designed to check the state of self-service, printer server and hardware services applications over a certain period of time, and to instruct the system on the correct algorithm for restarting these services.

There are two levels of functionality provided with the kiosk system: regular user functionality and administrator functionality. Normal operation of the system is managed

under general user functionality when the end user is designed to have access only to the self-service kiosk application and the operations it exposes.

The administrator functionality is available to the employees of the establishment that is using the kiosk system in its business processes and to the IT and technical maintenance service provider. This set of functions additionally covers the operations required to ensure the availability of the service on a day-to-day basis. The access to the administrator mode by the point-of-sale employees is managed by a special QR code that should be scanned by the barcode scanner. This QR code allows the employees to access the administrator menu allowing to perform the following operations:

- Collecting or refilling cash
- Reviewing and editing the price list
- Viewing the list of transactions
- Reprinting receipts or tickets
- Test printing
- Printing out fiscal reports for Lithuanian and Latvian installations
- Resetting API endpoint and session information
- Closing the self-service application
- Shutting down or rebooting the system

The QR code used to access the administrator mode is differentiated from the other QR codes used for identification of the regular products by adding a set of information containing predefined keywords and user identifier. The codes are static and vary based on the country and the company using them.

The representatives of the IT and technical maintenance provider company mostly manage the access to the system via TeamViewer application based on the set of Partner ID and password. If the maintenance requires physical presence of the provider's employees at the point of sale, the access to the machine can be also acquired via using the QR access code, and the manipulation of the system can be managed using the keyboard attached to the ports located inside the hardware case.

The updates of the system are also managed by the maintenance provider remotely and during the establishment's closed hours. For the duration of the update process the

technicians remotely manage disabling of the input from the local system and blackening the screen to prevent potential unauthorized access threat.

At the moment the new generation of software is being designed for the self-service kiosk system, that will require a different software service structure. In the new software setup, the single UWP kiosk application will be replaced by the combination of a React-based web application running inside the Electron shell and a local API that will be based either on REST (Representational State Transfer), gRPC or GraphQL approach communicating between themselves over HTTPS. This approach would allow for more flexibility when securing the system since the UWP technology used previously was not compatible with most of the existing kiosk security software solutions available on the market due to it being designed to function in an isolated AppContainer. Thus, the security for the current kiosk application is managed by the means of the application itself and restrictions placed on the operating system.

As there is a potential for the WCF technology to become deprecated with the evolution of .NET framework to .NET 5.0 there was a necessity to reconsider the way the hardware services and print server application should be implemented in the new setup. Thus, the functionality currently provided via the combination of these two services is going to be included into the local API for the next generation software design.

The new software setup would also introduce an SQLite database to be used for local storage. Initially it is supposed to act as a cache as well as to store state history information, external API access credentials and session information in order to allow the applications to come up without additional manual intervention after the service or computer is restarted.

The logging for the new system is planned to be cloud-based and organized using a .NET implementation of OpenTelemetry, a collection of tools, APIs, and SDKs (Software Development Kits) designed to instrument, generate, collect, and export telemetry data (metrics, logs, and traces) for analysis in order to understand software's performance and behaviour [14]. With this approach log records from all the devices included in the kiosk system will be aggregated at the same source allowing for easier access, improved control over access rights, and monitoring of the all the system components from one source.

The licensed version of Windows OS is supplied along with the ELO touchscreen computers used in the kiosk setups. Overall, the licensing process for the software used in the organization is fully outsourced to the technical and IT maintenance company.

3 Risk assessment

Risk assessment is one of the fundamental components of an organizational risk management process. Risk assessments are used to identify, estimate, and prioritize risk to organizational operations, assets, individuals, or other organizations, resulting from the operation and use of information systems. The purpose of conducting risk assessments is to inform decision makers and support risk responses by identifying relevant threats, vulnerabilities both internal and external to organization, impact to organization that may occur given the potential for threats exploiting vulnerabilities, and likelihood that harm will occur. The end result is a determination of risk, which is typically defined as a function of the degree of harm and likelihood of harm occurring [15].

The fact that the scope of the security analysis for this paper is limited to the kiosk machine itself allows to conduct an asset-based risk assessment for the setup. This approach is widely regarded as best practice as it presents a thorough and comprehensive approach to conducting risk assessments [16]. The asset-based risk assessment also has much stronger methodological basis than other alternative approaches, such as process-based risk assessment. The main reason for this is that this approach has been implemented by many of the organizations adopting ISO/IEC 27001 standard since it was a requirement before the revision of the standard in 2013 and has remained in the standard as one of recommended approaches after the revision [17].

Asset-based risk assessment process can be conventionally divided into four separate stages:

- Identification of the list of assets
- Identification of potential threats and vulnerabilities to the assets
- Risk probability and impact assessment
- Risk calculation

The result of the steps described above should be a prioritized list of risks in relation to the set of assets involved in the setup under research.

3.1 Identification of assets, threats, and vulnerabilities

In order to identify the assets for the risk assessment stage the whole set of assets can be divided into six categories: hardware, software, information, infrastructure, people, and outsourced services [18]. The classification of assets related to the security of the currently used semi-attended self-service kiosk setup is described in the Table 1.

Table 1. The classification of assets related to the self-service kiosk setup used in the organization. Abbreviations used: LV (Latvia), LT (Lithuania), EE (Estonia), FI (Finland).

Asset category	**Asset name**	**Countries**
Hardware	Touchscreen computer	LV, LT, EE, FI
	Cash recycling system	LV, LT, EE
	Receipt printer	LV, LT, EE, FI
	Barcode reader	LV, LT, EE, FI
	Unattended payment terminal system	LV, LT, EE, FI
	Fiscal device	LV, LT
	Network cabling	LV, LT, EE, FI
	Electrical cabling	LV, LT, EE, FI
Software	Operating system	LV, LT, EE, FI
	Kiosk application	LV, LT, EE, FI
	Hardware services	LV, LT, EE, FI
	Print server application	LV, LT, EE, FI
	Remote system access application	LV, LT, EE, FI
	System monitoring software	LV, LT, EE, FI
Information	System access QR codes	LV, LT, EE
	TeamViewer access credentials	LV, LT, EE, FI
	Locally stored logs	LV, LT, EE, FI
	Promotion codes	LV, LT, EE, FI
	External API access credentials	LV, LT, EE, FI
	Fiscal data	LV, LT
Infrastructure	Electrical supply	LV, LT, EE, FI
	Network connection	LV, LT, EE, FI
	Hardware case	LV, LT, EE, FI
	Premises	LV, LT, EE, FI

Asset category	Asset name	Countries
People	POS administrator	LV, LT, EE, FI
	POS staff	LV, LT, EE, FI
	Technical and IT maintenance specialists	LV, LT, EE, FI
Outsourced services	Technical and IT maintenance	LV, LT, EE, FI
	Payment terminal system setup and maintenance	LV, LT, EE, FI

The set of assets for the new kiosk setup will differ from the current one only by the contents of software section. For the new generation of kiosk software, the section will include the following assets:

- Operating system
- Web-based UI application powered by Electron JS
- Local API
- Local database
- Remote system access application
- System monitoring software

Having the list of main assets related to the setup identified allows to proceed to the next stage of identifying threats and vulnerabilities for these assets. The understanding of the difference between these two terms is crucial for conducting the risk assessment correctly. A threat is defined as any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. A vulnerability in its turn is a weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source [19].

The list of threats and vulnerabilities would be drawn with the consideration for both new and current kiosk setups in order to prevent future issues with the new components and avoid current vulnerabilities in the new system design. For the sake of brevity the risks that already have the controls described in the previous chapters implemented will be omitted in this assessment, and only untreated threats and vulnerabilities will be covered.

Since the focus of this paper is at the actual self-service kiosk solution, the remote Web API security is out of the scope of this research.

The list of threats and vulnerabilities related to the current setup used in the organization is represented in the Table 2.

Table 2. List of threats and vulnerabilities of the current self-service kiosk setup used in the organization

No	Threat	Vulnerability	Assets affected
1	Access to the operating system by unauthorized individuals	QR access codes are static and shared between all the staff of the venues belonging to one business within a country	Operating system
		Windows keyboard exposed in the normal operation of the self-service kiosk software allows to access Command Prompt and gain access to the system capabilities for installations in Estonia	Operating system
		There's no special system user with stripped privileges used for running the kiosk application for Estonian installations	Operating system
		System pop-up messages are not explicitly disabled for Finnish and Estonian installations	Operating system
		Windows system key combinations are not disabled	Operating system
		The analogue of mouse right-click for tablet is not explicitly disabled allowing to enter context menu for Finnish and Estonian installations	Operating system
2	Kiosk remaining in "Out of order" state for undefined amount of time due to software failure or malfunction of equipment	No visual or sound alerts are included with the physical setup to indicate improper system operation	Kiosk application
		There is no mechanism for identifying that the application is in hanging state	Kiosk application
3	Loss of electricity	There's no alternative electricity source available	Electrical supply

No	Threat	Vulnerability	Assets affected
4	Vandalism	Receipt printer slit is not protected against insertion of random items for the installations used in Estonia, Latvia, and Lithuania	Receipt printer
		Card reader slit is not protected against insertion of random items	Unattended payment terminal system
5	Loss of network connection	There's no mechanism for ensuring offline operation	Unattended payment terminal system, Kiosk application
		There's no offline alerting mechanism implemented	System monitoring software
6	Misuse of allocated system privilege set	There is no clear separation between the sets of privileges available for POS staff and technical maintenance specialists	Operating system
		There are unnecessary privileges available in the administrator menu for the POS staff	Operating system, Kiosk application
7	PIN theft	Plastic keys are used at the PIN pad for the installations in Finland allowing to read thermal signature	Unattended payment terminal system
8	Data theft or altering	There's no software or policy disallowing external devices included with the setup (vulnerability related to the internal staff)	Locally stored logs
		Windows keyboard exposed in the normal operation of the self-service kiosk software allows to view the clipboard history and get access to previously used inputs for Estonian installations	Promotion codes, External API access credentials
9	Exposure of the access credentials by employees having access to them	Employees having access to the sensitive information may not be fully educated on how to treat it securely	System access QR codes, Operating system, POS administrator, POS staff

No	Threat	Vulnerability	Assets affected
10	Unauthorized installation of software	Windows Developer Mode is enabled on the devices allowing non-certified applications to be executed for installations in Estonia and Finland	Operating system
		Microsoft Defender SmartScreen is disabled	Operating system
		Windows UAC is disabled	Operating system
11	Zero-day attacks	Automatic updates for Windows 10 are disabled, and all the updates including security ones are performed manually during the closed hours	Operating system
		There is no framework update policy implemented	Kiosk application, Print server application, Hardware services
12	Unauthorized modification of system settings	Windows UAC is disabled	Operating system

With the new generation of software incorporating a web-based UI application using Electron JS framework, there are additional risks arising concerning threats related to web application security that are described in the Table 3.

Table 3. List of threats and vulnerabilities related to the new generation of self-service software

No	Threat	Vulnerability	Assets affected
1	Cross-site scripting attack, Code injection attack, OS command injection attack	Electron framework is powered by Node.js which allows it to execute system affecting commands	Operating system, Web-based UI application
		Web-based applications allow referencing external untrusted resources	Operating system, Web-based UI application
		Electron framework allows backdoor injection via altering electron.asar file without changing the end application signature	Operating system, Web-based UI application

No	Threat	Vulnerability	Assets affected
2	SQL injection attack	User inputs provided in UI application are not validated	Operating system, Web-based UI application, SQLite database
		User inputs provided via UI application are not sanitized	Operating system, Web-based UI application, SQLite database

The threats and vulnerabilities listed above should be considered as an addition to the list related to the current self-service kiosk setup in order to accommodate for them when designing the new software system.

3.2 Risk probability and impact assessment

There are two main approaches to risk analysis: quantitative and qualitative. Quantitative risk analysis uses available relevant and verifiable data to produce a numerical value which is then used to predict the probability (and hence, acceptability) of a risk event outcome. Qualitative risk analysis, on the other hand, applies subjective assessment of risk occurrence likelihood (probability) against the potential severity of the risk impact to determine the overall severity of a risk [20].

It is commonly assumed that the quantitative risk analysis by the nature of it should provide more useful results, however there is a set of limitations to it that make the qualitative approach a better choice. First of all, in order to perform quantitative analysis properly there should be a large set of statistical data from previous periods available for evaluation of likelihood. Moreover, it requires the damage that may be caused by risk realization to be measurable, which is not always straightforward as the risk may affect multiple systems at once and to different extent depending on the response time.

Since the organization under research does not have precise statistical data about the incidents regarding the self-service kiosk setup, and due to the fact that the evaluation of the effect caused by risks might vary based on the number of installations affected by the vulnerability being exploited, the risk analysis will be carried using the qualitative

method. This approach also allows to analyse both existing and future setups using the same scale and prioritize the risks identified.

There can be two approaches distinguished within qualitative risk analysis: simple and detailed risk assessment. In simple risk assessment the consequences and the likelihood are assessed directly – once the risks are identified, pre-defined scales are used to assess separately the consequences and the likelihood of each risk. In the detailed risk assessment, instead of assessing consequences and likelihood, three elements are assessed: asset value, threat, and vulnerability. Using these three elements in detailed risk assessment allows to indirectly assess the consequences and likelihood: assessing the asset value is assessing which kind of damage or consequence could happen to this asset if its confidentiality, integrity, or availability is endangered; both threats and vulnerabilities directly influence the likelihood – the higher the threat and the higher the vulnerability, the more likely the risk will happen, and vice versa [21].

Using the detailed risk assessment would allow to consider the value of the asset, as well as to give a correct evaluation of the threat and vulnerability impact, as in particular cases the threat itself might have high probability, however the vulnerability allowing to exploit it might be a minor one.

In order to balance the weight of consequence and possibility assessment for the detailed method the scales used to evaluate threats and vulnerabilities have the range that is twice smaller than for the scale used to measure asset value. For the assessment of the risks related to the self-service kiosk setup used in the organization the following scales will be used:

- Scale from 0 to 4 for asset value assessment, with the following range of values: 0 - Very low, 1 - Low, 2 - Medium, 3 - High, and 4 - Very high
- Scale from 0 to 2 for threat assessment, with the following range of values: 0 - Low, 1 - Medium, and 2 - High
- Scale from 0 to 2 for vulnerability assessment, with the following range of values: 0 - Low, 1 - Medium, and 2 - High

The ranges for the scales above have been defined as quite narrow ones since there is not much variation in the asset value due the limited amount of assets studied for the setup.

The probability and risk assessment for the current self-service kiosk setup is presented in the Appendix 2 Risk calculation for the current self-service kiosk setup used in the organization, and for the additional threats and vulnerabilities identified for the new generation of self-service software in the Appendix 3 Risk calculation for the new generation of self-service software.

3.3 Risk calculation

There are two methods that are commonly used for risk calculation that is based on qualitative method. According to the first method, the values received for a single asset-threat-vulnerability combination based on the assessment scales defined need to be added to receive total risk calculation; for the second method, the same values need to be multiplied between themselves. [21] The main difference between these approaches comes from the range of total values received which is greater for the multiplication formula and can allow for more detailed prioritizing. However, there is a drawback to this approach, as in case when asset value for the combination is evaluated as Very low, or either threat or vulnerability level is evaluated as Low, the total risk value will become equal to zero, and the risk will most likely be neglected. Therefore, for the purpose of this paper the sum-based method is going to be used.

The calculation of the final risk values for threat-vulnerability-asset combinations is carried based on the assessments made for these elements in the previous part of the paper. The calculated risk values are represented in the Appendix 2 Risk calculation for the current self-service kiosk setup used in the organization for the risks related to the currently used self-service kiosk setup, and for the new self-service software setup in the Appendix 3 Risk calculation for the new generation of self-service software.

From the results of risk calculation, it was agreed within the organization that the acceptable level of risk is 6. All the risks with the total calculated value of 7 and higher need to be treated in prioritized way. The vulnerabilities that require controls to be implemented based on the calculation made are the following (listed in prioritized order):

- There's no special system user with stripped privileges used for running the kiosk application for Estonian installations

- Windows keyboard exposed in the normal operation of the self-service kiosk software allows to access Command Prompt and gain access to the system capabilities for installations in Estonia
- Windows keyboard exposed in the normal operation of the self-service kiosk software allows to view the clipboard history and get access to previously used inputs for Estonian installations in relation to confidentiality of external API access credentials
- Electron framework allows backdoor injection via altering electron.asar file without changing the end application signature in relation to confidentiality, integrity and availability of operating system and related data
- Electron framework is powered by Node.js which allows it to execute system affecting commands
- Windows system key combinations are not disabled
- QR access codes are static and shared between all the staff of the venues belonging to one business within a country

4 Design of risk treatment proposal

After the identification of the set of vulnerabilities posing unacceptable risks to the self-service kiosk setup, it is required to design a set of controls to be implemented in order to treat these vulnerabilities. This part of the paper will introduce the proposed solutions to treatment of the risks identified that will allow to enhance the security of the semi-attended self-service kiosk setup used in the organization.

4.1 Implementation of TOTP (Time-based One-Time Password) for accessing kiosk administrator mode

The access to the administrator mode for the self-service kiosk setup used in the organization is currently managed using a set of static QR codes, that are shared between all the machines installed at the venues belonging to a single business. This arrangement is exposing the setup to the threat of intentional or unintentional disclosure of the code to unauthorized individuals, and them using it to access the kiosk system capabilities. It is especially detrimental since there currently is no procedure in place for changing these access codes easily.

In order to address this vulnerability implementing TOTP for getting access to administrator mode can be used. The TOTP (Time-based One-Time Password) is a term that refers to a computer algorithm that generates a one-time password which uses the current time as a source of uniqueness, or the resulting one-time password itself [22]. The TOTP is usually valid for a short period of time (commonly varying between 30 seconds and 1 minute), and can be used only once to authenticate the user with the system.

For the setup in question it would still be most beneficial to use the authentication in the form of QR code, as it would limit the need for user input via the kiosk screen. Setting up a system with passwords or passcodes being typed in using the touch screen would in its turn present an additional security concern, as it would require designing a separate view within the application to provide functionality for putting in the administrator mode credentials, as well as creating a set of new input fields on the public system that would need to be secured separately.

However, using TOTP-based QR codes comes with additional considerations, such as a requirement to restrict access to the access code generation mechanism to authorized individuals, and a need to determine a way to differentiate between authorization QR codes and the codes used for regular products registered with the kiosk system.

The suggested implementation for this solution would be to design a custom mobile application that will be installed on the corporate mobile devices allowing authorized employees to generate TOTP-based QR codes for getting access to kiosk application administrator menu.

The proposed authentication mechanism to be used is similar to the Google Authenticator application mechanism, however utilizing only the TOTP option, using a different hashing algorithm to provide for better performance and security, and generating QR codes instead of numeric passwords. The access to the mobile application will be account-based and managed by a combination of username and password, which will be unique per employee. The credentials required to access the application will be shared personally to every employee, and disabled in case employee resigns from the company. The internal logics of the application should have a limited set of capabilities for logged in employees, allowing the following actions:

- Changing employee's personal password
- Adding a new managed self-service kiosk machine to the account
- Selecting a kiosk machine to get access to
- Generating a TOTP to access the selected self-service setup

As this authenticator application is designed as an additional external feature to ensure the secure access to the self-service system, it will use the same remote database as the main application, but with a limited set of rights.

The initial registration of employees with the system will be managed by adding their credentials to the main external API database manually. Every employee registered will be assigned a Nano ID identifier, a GUID (Globally Unique Identifier), a set of self-service setups that the employee should be granted access to, and a status of employee record. There also should be two different roles defined for POS employees: an administrative role for the POS administrator, and a regular user role for the rest of POS staff that is supposed to have access to the self-service system.

To allow employees to use the mobile application for authenticating with the self-service setup the shared secret should be generated for each employee. This would require defining an additional algorithm within the existing kiosk application, that would import the information related to the current installation from the external API database and store it locally using an encrypted SQLite database as a storage, display the list of users imported, and allow to generate and reset user secrets. Based on the user information imported, a secret should be generated for each employee using one out of two scenarios: for the employees in administrator role the secret should be generated under the supervision of a technical and IT maintenance specialist; for regular POS staff members the procedure to generate the secret should be initiated by the POS administrator already registered with the system.

Setting up the authentication system for the first time for each employee should involve the following steps:

- The user for whom the secret key is required to be generated (further referred to within this list as the Employee) needs to log into the mobile authenticator application using his personal set of credentials
- The Employee should change the initial one-time password to the custom one to ensure, that the current value is known only to one person, and log into the application again using the new set of credentials. There should be restrictions set for the custom user password to be at least 10 symbols long and to contain at least one upper and lowercase alphabetic character, at least one digit and one special symbol
- Upon logging into the authenticator application with the new credentials, the employee should select the self-service setup to connect to from the list of setups that are available for the account, and select the option to generate a secret key, which will be displayed on the screen of the mobile device in the form of QR code
- At the self-service machine the person initiating the procedure should select the Employee from the list of users imported from the database for a particular self-service setup: for the maintenance specialists this list should include all the employee records imported, while for administrator users it should only include regular POS employees

- The Employee should scan the QR code key generated on the mobile device with the barcode scanner at the kiosk setup, and wait for the self-service system to confirm the success of the operation

The schematic diagram of the user registration process using the mobile authenticator application is represented in the Figure 3.

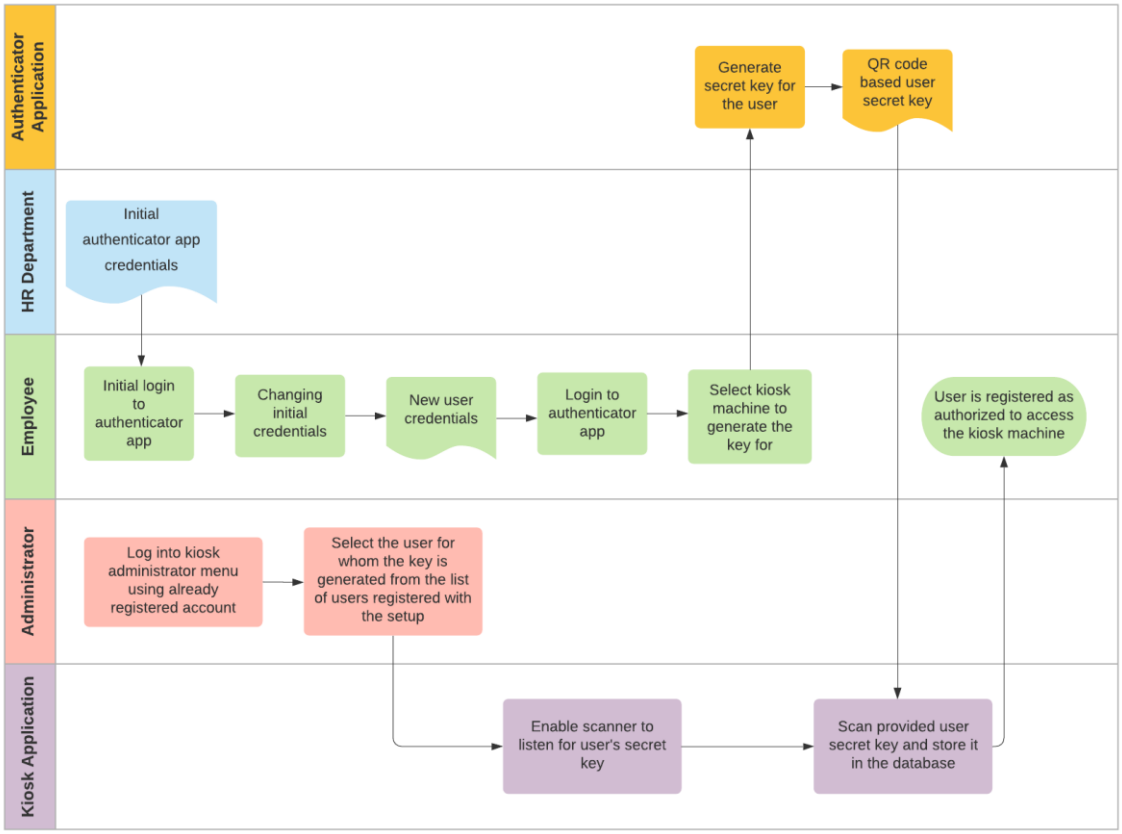

Figure 3. User registration flow using the mobile authenticator application

After the shared key has been stored with both self-service kiosk and mobile application account, it can be used to generate TOTP-based QR codes for accessing the administrator menu of the kiosk application.

The secret key used for authentication will be generated according to NIST (National Institute of Standards and Technology) recommendations for symmetric key generation [23] using the capabilities of FIPS 140-2 (Federal Information Processing Standard 140-2) compliant .NET RNGCryptoServiceProvider class [24]. This class implements a CSPRNG (Cryptographically Secure Pseudorandom Number Generator) using the implementation provided by the cryptographic service provider [25].

According to NIST recommendations, the formula for generating symmetric cryptographic keys using the output of CSPRNG should look as described in the Equation (1) below, where B denotes the resulting symmetric key, and U and V values are bit strings of the same length as the resulting key, for which the value of V should be determined in a manner that is independent of the value of U [23]. The $\oplus$ sign in the Equation (1) denotes a XOR (Exclusive OR) operation.

$$B = U \oplus V \tag{1}$$

In order to create secret keys that are specific to the users and do not rely exclusively on the randomness of the output of the CSPRNG used, the GUID assigned to every user upon registration in the database should be used as the V value in the Equation (1). As GUID produces a 128-bit output it is reasonable to implement a 128-bit key.

In order to generate the TOTP for accessing the kiosk setup administrator menu that will be valid for 60 seconds there are two main components required: the secret key for the account and the Unix timestamp that is the higher closest multiple of 60 seconds to the current timestamp. These components need to be concatenated and passed to a hash function in order to produce a TOTP. The hash function proposed for this operation is KangarooTwelve eXtendable Output Function (XOF), a generalization of a hash function that can return an output of arbitrary length. KangarooTwelve is based on a Keccak-p permutation specified in FIPS 202 [26] and has a higher speed than SHAKE (Secure Hash Algorithm and Keccak) and SHA-3 (Secure Hash Algorithm version 3) algorithms. It also can provide 128-bit security strength with the output that is at least 128 bit long for 128-bit second preimage security, and with the output at least 256 bits long for 128-bit collision security [27], which matches the length required for 128-bit security for the SHA-3 [28].

As an input the KangarooTwelve algorithm requires a message M, an optional customization string C commonly used to identify a domain, both byte strings of variable length, and a positive integer value L to indicate the length for the output in bits [29]. For the current implementation, as the secret keys will vary between the kiosk setups, there is no need to implement the customization string. The message M for the function will be represented by the result of concatenation of the user secret key and the timestamp in the

corresponding order, and the length of the output will be set as 256 bits to provide for both 128-bit collision and second preimage security.

After a hash is generated using the KangarooTwelve algorithm, the Nano ID user identifier in the form of a bit string should be appended to it in a non-encrypted way in order to provide the recipient system with the means to identify the secret key to be used within a reasonable amount of time. Then the final byte string should be converted into string format and a QR code should be generated from the resulting string.

The process of generating the user secret key and kiosk access key is represented at the Figure 4.

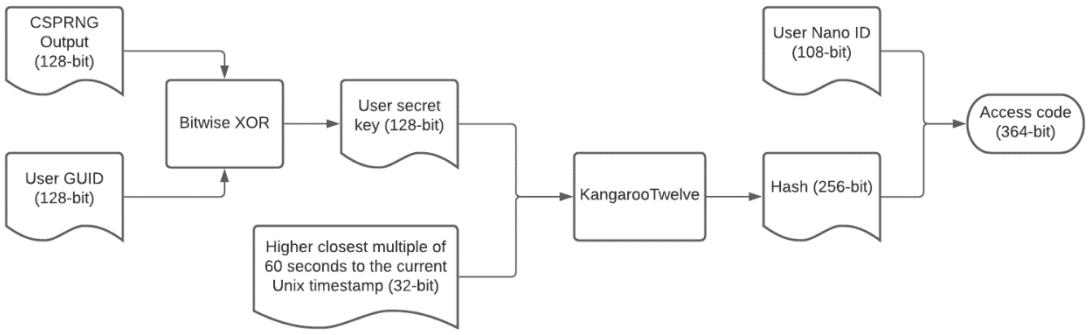

Figure 4. User secret key and kiosk access key generation algorithm

In order to generate the TOTP an employee should log into his account in mobile application, select a kiosk machine out of the list of machines connected to the account, and click “Generate access code” button. The QR code will be displayed on the mobile device screen along with the countdown timer displaying the remaining validity time for the code.

The kiosk application should be configured to listen for the access QR codes in the background the same way as it is currently listening for the static code. Once the code is scanned it should be validated against the predefined bit length for access codes equal to 364 bits. If that condition applies, the code should be split into 2 parts: the first part of 256 bits containing the access code, and 108-bit string containing user’s Nano ID. The application needs to validate that the Nano ID is a string containing only alphanumeric symbols, search for the corresponding user secret key based on the Nano ID provided, and generate a set of three hashes using the same algorithm as is used for the generation

of the access hash on the mobile device. However, to accommodate for the time lag between access code generation and scanning moments, and for a potential time difference between the devices, three different timestamps should be used: the higher closest multiple of 60 seconds to the current timestamp and two closest multiples of 60 seconds (higher one and lower one) to the first timestamp calculated. The results of the hash function using the combinations of secret user key with these timestamps should be compared against the first 256 bits of the QR code scanned, and in case they match, the user should be presented with the administrator mode kiosk application menu. This flow is described in the diagram in the Figure 5.

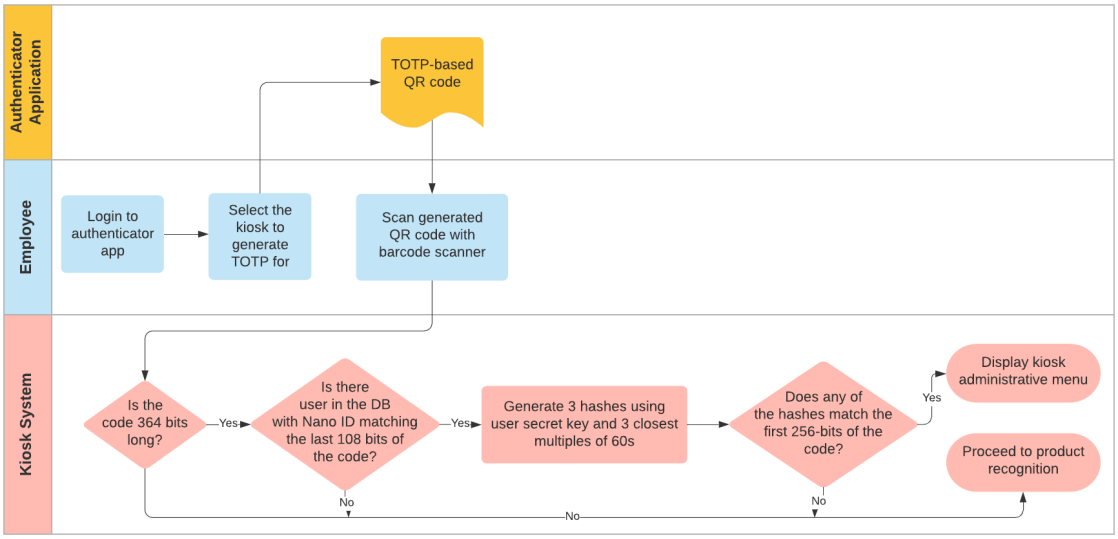

Figure 5. User flow for accessing the kiosk administrator menu using the authenticator application

As there are two different groups of employees that require such access, the role of employee who was identified by the access code should be checked from the database, and the corresponding capabilities should be displayed within the application menu. If none of the results of the hash function has matched the received hash, the “Product not found” error screen should be displayed by the kiosk application, and the information regarding the attempt should be logged with the Warning log level, including the data read from the QR code and the description of the operation performed.

In order to prevent other employees from using the same account with the mobile application, there should be a session duration parameter implemented that would log the employee out of the account after 5 minutes of inactivity. The administrator mode kiosk

menu is designed to be exited from manually, however there should also be an inactivity timer implemented for exiting the administrator menu after 1 minute of inactivity.

Removing of employees' secret keys with this setup can be managed remotely via marking the user record as disabled in the external API database, that will be exported regularly over fixed time intervals to the kiosk setup to update local user list. The export should be also available to be performed manually for critical cases, when the secret key should be disabled urgently. For forcing the reset of the user secret, the user record should be marked with reset required status, which will disallow the secret key stored locally to be recognized by the setup. Resetting the key should follow the same procedure as the initial key generation, and shall be initiated by the corresponding supervising person.

For the technical and IT maintenance service specialists, an additional maintainer role needs to be defined in the user roles table. The initial creation of the maintainer secret keys should be managed differently from those for POS employees, as they need to be distributed to all the kiosk machines the service is provided for. Therefore, the keys should be generated for every maintainer account in a centralized way and exported to all the serviced setups along with the rest of the user data in an encrypted way. The application menu available to the maintenance specialists should additionally include the option to exit the kiosk application, which would allow to access the operating system.

4.2 Implementation of a system of access rights levels

This part is designed as a control for the absence of special system user with stripped privileges for running the kiosk application vulnerability identified in Estonian kiosk installations. However, this solution will also address the threat of misuse of allocated system privilege set and the vulnerabilities it explores, along with the vulnerability of having no software or policy disallowing connecting external devices included with the setup.

Due to the fact, that there are several groups of users that require access to the self-service kiosk setup it is reasonable to introduce a system that will differentiate between access right sets for these user groups. According to the needs identified while interviewing the parties that are dealing with the setup from the organizational side, the following user access rights sets can be implemented:

- Regular user access rights (regular user mode)
- Manager access rights (administrator mode)
- Maintainer access rights (maintainer mode)

The separation between categories does not directly imply the different sets of Windows system user privileges, but rather defines a combination of system restrictions and self-service application functions available for each group.

The set of regular user access rights in this classification is designed to provide service to the end customer, but restrict the undesired operations. These rights should ensure that the regular user is capable to use the self-service kiosk application and have access to all of its capabilities, but is not allowed to access any other system functions. To achieve that, a special Windows system user should be defined for running the self-service application, with the following capabilities and features disabled:

- Windows on-screen keyboard
- System pop-up messages
- Windows shortcut key combinations
- The analogue for mouse right-click for Windows tablet mode
- Automatic updates for Windows
- Access to Control Panel
- Access to Command Prompt
- Access to Windows Registry editing tools
- Clipboard history
- Windows AutoPlay feature for CDs, DVDs, BDs, USB sticks and memory cards

Additionally it is viable to define a fixed list of applications that are allowed to run using Windows 10 Pro Local Group Policy Editor to allow only the user installed software that is required to run, monitor and manage the self-service kiosk.

The regular user should be able to perform the following operations within the kiosk application: log into user account with the ID card for the businesses supporting such functionality, select items from the application menu, scan products using barcode scanner, scan promotion codes and order invoices from the cinema using barcode scanner, select shows and tickets for installations used in cinemas, redeem reservations, print out previously purchased tickets, and pay for the selected or scanned items.

The manager access rights are supposed to cover the set of permissions required for the POS administrator and staff members in order to ensure the proper operation of the kiosk setup. Current set of operations available to the employees at the POS is allowing a wider range of capabilities, than is actually required for maintaining normal operation of the software. The proposed solution should limit the system rights available to the POS employees by using the same system user account as is used for the regular users. The difference in access rights should come from the application functionality available for the employees via the application administrator menu.

Additional capabilities exposed by the application to all POS employees should include the following:

- Collecting or refilling cash
- Reprinting receipts or tickets
- Test printing
- Printing out fiscal reports for Lithuanian and Latvian installations, managed via a select field with predefined values allowing to choose the type of report, and the date picker field in form of calendar to choose the range of dates for report generation
- Resetting API endpoint and session information
- Shutting down or reloading the system

Access to the operating system for the manager access rights group should also be restricted, and all of the capabilities listed above should be managed using predefined commands assigned to application functions, with minimum user input allowed to be provided via a set of fields containing predefined values in order to eliminate the need for arbitrary textual inputs. The options to review and edit the price lists and to view the list of transactions for the kiosk machine previously available through the self-service application menu should be set to be managed from the salespoint attended device instead.

The access to the manager mode should be managed using TOTP-based QR codes generated using the authenticator application described in the previous part of the paper. For the POS administrator the self-service application administrator menu should additionally provide an option to set a secret access key for the regular POS employees. Selecting this option should display the list of employees that are allowed access to the

current machine and can be selected for assigning a generated key to their accounts. The same option shall be used in case when resetting the secret system access key is required.

The maintainer set of access rights should be shared only to the technical and IT maintenance company representatives in order to allow them to monitor and manage the system. There should be a separate Windows system administrator account created for these purposes, and the set of access credentials to it should be shared only to the maintenance company.

The access to the maintainer mode should be possible both remotely using the TeamViewer application, and locally. For accessing the setup locally the maintenance specialist should log into the corresponding user account using the custom mobile authenticator application described in the previous chapter with the individual set of credentials and generate a TOTP-based QR code to be scanned with the barcode scanner. For maintainer role the administrator menu of the self-service kiosk application that is accessed with the QR code should contain an additional function for exiting the application to the Windows desktop. The option to set a secret access key for the self-service setup in the maintainer mode should also display the POS administrator accounts along with the regular employees list.

The Windows system administrator account can then be accessed by inputting the administrator credentials using a hardware keyboard connected to the USB port available within the hardware case, that can be unlocked using the physical keys that are also shared to maintenance specialists.

For the remote connection via TeamViewer there is an inbuilt capability to register to the administrator account using the Windows Authentication capabilities. For this, the TeamViewer application should be ran under administrator privileges on the remote machine, and the administrator credentials should be entered via the Advanced settings TeamViewer Authentication menu.

4.3 Implementation of QR code based promotion codes

The solution described in this part of the thesis is designed to address the vulnerabilities posed by the use of Windows keyboard within the publicly available self-service kiosk application, such as Windows keyboard being exposed in the normal operation of the self-

service kiosk software allowing to access Command Prompt and gain access to the system capabilities, and to view the clipboard history and get access to previously used inputs for installations in Estonia.

The need for having user inputs in textual form within the kiosk application is defined only by the requirement for the system to be able to accept promotion codes distributed to the end clients as a part of marketing strategy. Currently such codes are distributed to the clients using email messages and promotion flyers, and are used to provide users with discounts for purchases made via websites, attended points of sale, and self-service kiosks. While using the promotion codes with the website interface would still require it being in textual form in most of the cases to provide convenience to the users, for the rest of implementations the QR code representing the same textual value may be used. Therefore, the codes should be distributed in pairs representing them in both textual and QR code formats to accommodate for all usage options.

In order to implement this capability with the self-service kiosk application, the view prompting clients to input their promotion code and displaying the on-screen Windows keyboard should be replaced with the view allowing clients to scan the QR promotion code using the barcode scanner installed with the setup.

This approach would allow to eliminate the need for using Windows on-screen keyboard within the kiosk self-service application and remove the vulnerabilities connected to it without altering the business process and overall capabilities of the system.

The Windows on-screen keyboard therefore should be disabled for the regular system user with stripped rights using the Windows Registry Editor by setting the ShowTabletKeyboard registry option to zero.

4.4 Addressing Electron JS framework vulnerabilities

The introduction of the new generation software design for the self-service kiosk requires addressing additional vulnerabilities arising from the technologies used. There are two main vulnerabilities related to the Electron JS framework used for running the web-based UI application that have been evaluated in the combination with potential threats and assets as causing unacceptable risk levels: the framework is powered by Node.js which

allows it to execute system affecting commands, and it also allows backdoor injection via altering electron.asar file without changing the end application signature.

The electron.asar file is a part of Electron JS installation resources that contains the boot-loader responsible for the preparation of the Chromium browser environment allowing to run Electron-powered applications. This file is stored in a non-encrypted, non-obfuscated way, so it is possible for an attacker that can gain access to the filesystem to modify this file while not modifying the signature of the actual executable file. The modifications made to the file may allow for data exfiltration, injection of keylogging capabilities, usage of inbuilt camera on the device, or taking screenshots of the device's screen.

In order to address the vulnerability related to the electron.asar file, the access to the part of the filesystem storing the installation folder should be restricted for the regular user. To do so, it is recommended to install the application as a high privileged administrator user, and run it under the user with the regular set of privileges. Frequent updates issued to the application also address this problem, as the electron.asar file is getting overwritten on every software update.

Due to Electron JS being a framework that allows building desktop applications using web technologies, the applications using it are prone to the common web application vulnerabilities. As the framework uses Node.js as its engine, it is allowed to access the filesystem and the user shell of the machine the application is running on, so various injection attacks such as cross-site scripting and code injection can lead to more severe consequences than in the case of regular web applications. To address this threat the nodeIntegration parameter should be verified to be explicitly set to false for the applications to be deployed.

Along with disabling Node.js integration, it is recommended to enable Context isolation feature allowing to ensure that preload scripts and Electron's internal logic run in a separate context to the main web application. This is important for security purposes as it helps prevent the application from accessing Electron internals or the powerful APIs that preload script might need access to [30].

It is also crucial to use the latest version of Electron JS framework, as there are critical vulnerabilities within the previous versions allowing the malicious actors to re-enable the Node.js integration during the runtime [31].

5 Summary

The growth of the share of self-service based operations in the service industry overall has made self-service kiosk setups an attractive target for various kinds of attacks requiring organizations using them to pay more attention to the security of such setups. This thesis analyses the security of semi-attended self-service kiosk installations based on the example of the kiosk setups used by Apollo Digital. The paper provides a detailed description of the physical and software components of the kiosk installation along with the controls that are implemented within the organization to ensure security of these components. Based on the description a list of assets related to the kiosk setup was identified, along with associated threats and vulnerabilities. This list was used to calculate the total risk levels using qualitative risk assessment, and identify risks of the levels 7 and 8 on the scale from 0 to 8 as unacceptable and therefore requiring prioritized treatment. Most of these risks are caused by existing vulnerabilities related to allocation of operating system and application privileges, as well as to some inherent vulnerabilities of the Electron JS framework to be used in the new software design for the kiosk application.

The solutions for risk treatment proposed by this thesis include the following:

- Development of the TOTP-based QR code authenticator application that would allow to enhance the security of the method for accessing the management and maintenance kiosk system capabilities
- Splitting the users of the self-service kiosk setup into different privilege groups with the limitations set according to the needs of each group
- Replacing textual user input capability for the kiosk application by the usage of QR codes
- Security configurations for development of the new generation web-based UI application utilizing the Electron JS framework

Implementation of solutions proposed would allow the organization to enhance the security of the currently used self-service kiosk system, as well as to address the vulnerabilities of the new design of kiosk setup.

References

[1] KMA (Kiosk Manufacturer Association). (2020, August 31). Kiosk ADA Guidelines for Accessibility - Hardware and Software. Kiosk Manufacturer Association. Retrieved May 11, 2021 from https://kma.global/guidelines-kiosk-ada-accessibility/

[2] Office for Civil Rights (OCR). (2017, June 16). HIPAA for Professionals. U.S. Department of Health & Human Services. Retrieved May 11, 2021 from https://www.hhs.gov/hipaa/for-professionals/index.html

[3] PCI Security Standards Council LLC. (n.d.). About Us. Official PCI Security Standards Council Site. Retrieved April 9, 2021 from https://www.pcisecuritystandards.org/about_us/

[4] The University of Texas Rio Grande Valley (UTRGV) Information Security Office. (2018, November). The University of Texas Rio Grande Valley (UTRGV) Kiosk Security Standard (No. 1.0). UTRGV. Retrieved May 11, 2021 from https://www.utrgv.edu/is/_files/documents/kiosk-security-standard.pdf

[5] The UC (University of California) Berkeley Information Security Office. (n.d.). Campus Guidelines for Kiosk Workstations | Information Security Office. The UC (University of California) Berkeley. Retrieved May 11, 2021, from https://security.berkeley.edu/campus-guidelines-kiosk-workstations

[6] Provisio Software Engineering. (n.d.). Kiosk Software - SiteKioskTM for Windows. Retrieved May 11, 2021, from https://www.provisio.com/web/us/products/windows-kiosk-software-sitekiosk

[7] KioWare. (n.d.). Windows Kiosk Software. KioWare - Kiosk System Software. Retrieved May 11, 2021, from https://m.kioware.com/windows

[8] Lindsay, G. (2019, January 9). Set up a single-app kiosk (Windows 10) - Configure Windows. Microsoft Docs. https://docs.microsoft.com/en-us/windows/configuration/kiosk-single-app

[9] Ingenico Group. (2017, July). iUC250 PCI PTS security policy (ICO-OPE-02546-EN-V2). PCI Security Standards Council. Retrieved April 8, 2021, from https://www.pcisecuritystandards.org/ptsdocs/4-30164_ICO-OPE-02546-EN-V2_PCI_PTS_Security_Policy_iUC250.pdf

[10] Ministry of Finance of the Republic of Lithuania. (2003). Order on the Rules for the Use of Cash Devices and Direct Communication Network Terminals and the Approval of the Form of the Decision to Register a Cash Device FR1156 (V-255). Register of Legal Acts of the Republic of Lithuania. Retrieved April 14, 2021 from https://www.e-tar.lt/portal/lt/legalAct/TAR.ACCBA203C552/asr

[11] Cabinet of Ministers of the Republic iof Latvia. (2014). Regulations Regarding Technical Requirements for Electronic Devices and Equipment for the Registration of Taxes and Other Payments (95). Legal Acts of the Republic of Latvia. Retrieved April 9, 2021 from https://likumi.lv/ta/en/en/id/265486-regulations-regarding-technical-requirements-for-electronic-devices-and-equipment-for-the-registration-of-taxes-and-other-payments

[12] Apriorit. (2019, October 29). AppContainers for Windows 8: What Are They and How Can You Create Them. Medium. Retrieved April 13, 2021 from https://medium.com/that-

feeling-when-it-is-compiler-fault/appcontainers-for-windows-8-what-are-they-and-how-can-you-create-them-e5970a28eea4

[13] F-Secure. (n.d.). About F-Secure Protection Service for Business (PSB). F-Secure User Guides. Retrieved April 14, 2021 from https://help.f-secure.com/product.html?business/psb-portal/latest/en/concept_F0611339D8E44CCC975A977F5428AA5D-psb-portal-latest-en

[14] The OpenTelemetry Authors. (2021, April 7). Documentation. OpenTelemetry. Retrieved April 22, 2021 from https://opentelemetry.io/docs/

[15] National Institute of Standards and Technology (NIST) Computer Security Division. (2012, September). NIST Special Publication: Guide for conducting risk assessments (No. 800–30R1). NIST. Retrieved April 15, 2021 from https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-30r1.pdf

[16] Warren, J. (2018, December 28). Conducting an asset-based risk assessment in ISO 27001:2013. IT Governance USA Blog. Retrieved April 15, 2021 from https://www.itgovernanceusa.com/blog/conducting-an-asset-based-risk-assessment-in-iso-270012013

[17] Kosutic, D. (n.d.). ISO 27001 risk assessment methodology – How to write it. 27001Academy. Retrieved April 15, 2021 from https://advisera.com/27001academy/knowledgebase/write-iso-27001-risk-assessment-methodology/

[18] Kosutic, D. (n.d.). ISO 27001 Asset Register: How to develop an asset inventory. 27001Academy. Retrieved April 15, 2021 from https://advisera.com/27001academy/knowledgebase/how-to-handle-asset-register-asset-inventory-according-to-iso-27001/

[19] National Institute of Standards and Technology (NIST) Computer Security Division. (2006, March). Federal Information Processing Standard (FIPS) Publication: Minimum Security Requirements for Federal Information and Information Systems (No. 200). NIST. Retrieved April 16, 2021 from https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.200.pdf

[20] Shuttleworth, M., & Shuttleworth, M. (2017, March 12). Qualitative vs. Quantitative Risk Analysis: What's the difference? Project Risk Manager. Retrieved April 20, 2021 from https://www.project-risk-manager.com/blog/qualitative-and-quantitative-risk-analysis/

[21] Kosutic, D. (n.d.). ISO 27001 risk analysis: How to assess consequences & likelihood. 27001Academy. Retrieved April 20, 2021 from https://advisera.com/27001academy/knowledgebase/how-to-assess-consequences-and-likelihood-in-iso-27001-risk-analysis/

[22] Internet Engineering Task Force (IETF). (2011, May). Request For Comments – TOTP: Time-Based One-Time Password Algorithm (No. 6238). Retrieved April 24, 2021 from https://tools.ietf.org/html/rfc6238

[23] Barker, E., Roginsky, A., & Davis, R. (2020, June). National Institute of Standards and Technology (NIST) Special Publication: Recommendation for Cryptographic Key Generation (No. 800-133R2). NIST. Retrieved April 26, 2021 from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf

[24] Microsoft. (2021, March 25). Federal Information Processing Standard (FIPS) 140 Validation - Windows security. Microsoft Docs. Retrieved April 26, 2021 from https://docs.microsoft.com/en-us/windows/security/threat-protection/fips-140-validation#_microsoft_fips_140

[25] Microsoft. (n.d.). RNGCryptoServiceProvider Class (System.Security.Cryptography). Microsoft Docs. Retrieved April 26, 2021 from https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.rngcryptoserviceprovider?redirectedfrom=MSDN&view=net-5.0

[26] National Institute of Standards and Technology (NIST). (2015, August). Federal Information Processing Standard (FIPS) Publication: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions (No. 202). NIST. Retrieved April 26, 2021 from https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

[27] Viguier, B. (2020, March). KangarooTwelve. Internet Engineering Task Force (IETF). Retrieved April 26, 2021 from https://tools.ietf.org/id/draft-irtf-cfrg-kangarootwelve-02.html

[28] Barker, E. (2020, May). National Institute of Standards and Technology (NIST) Recommendation for Key Management: Part 1 (No. 800-57PT1R5). NIST. Retrieved April 26, 2021 from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf

[29] Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., Van Keer, R., & Viguier, B. (n.d.). KangarooTwelve: fast hashing based on Keccak-p. Retrieved April 26, 2021 from https://keccak.team/files/KangarooTwelve.pdf

[30] The OpenJS Foundation. (n.d.). Electron Documentation: Context Isolation. Electron JS. Retrieved April 27, 2021 from https://www.electronjs.org/docs/tutorial/context-isolation

[31] National Institute of Standards and Technology (NIST). (2018, March 23). NVD - CVE-2018-1000136. National Vulnerability Database (NVD). Retrieved April 27, 2021 from https://nvd.nist.gov/vuln/detail/CVE-2018-1000136

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Darya Harachka

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Securing a Semi-Attended Self-Service Kiosk based on the Example of the Services offered by Apollo Digital", supervised by Priidu Paomets
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

17.05.2021

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – Risk calculation for the current self-service kiosk setup used in the organization

No	Threat	Vulnerability	Assets affected	Threat value (0-2)	Vulnerability value (0-2)	Asset value (0-4)	Risk value (0-8)
1	Access to the operating system by unauthorized individuals	QR access codes are static and shared between all the staff of the venues belonging to one business within a country	Operating system	2	1	4	7
		Windows keyboard exposed in the normal operation of the self-service kiosk software allows to access Command Prompt and gain access to the system capabilities for installations in Estonia	Operating system	2	2	4	8
		There's no special system user with stripped privileges used for running the kiosk application for Estonian installations	Operating system	2	2	4	8
		System pop-up messages are not explicitly disabled for Finnish and Estonian installations	Operating system	2	0	4	6
		Windows system key combinations are not disabled	Operating system	2	1	4	7

No	Threat	Vulnerability	Assets affected	Threat value (0-2)	Vulnerability value (0-2)	Asset value (0-4)	Risk value (0-8)
		The analogue of mouse right-click for tablet is not explicitly disabled allowing to enter context menu for Finnish and Estonian installations	Operating system	2	0	4	6
2	Kiosk remaining in "Out of order" state for undefined amount of time due to software failure or malfunction of equipment	No visual or sound alerts are included with the physical setup to indicate improper system operation	Kiosk application	1	1	3	5
		There is no mechanism for identifying that the application is in hanging state	Kiosk application	1	1	3	5
3	Loss of electricity	There's no alternative electricity source available	Electrical supply	1	1	3	5
4	Vandalism	Receipt printer slit is not protected against insertion of random items for the installations used in Estonia, Latvia, and Lithuania	Receipt printer	1	1	3	5
		Card reader slit is not protected against insertion of random items	Unattended payment terminal system	1	1	4	6

No	Threat	Vulnerability	Assets affected	Threat value (0-2)	Vulnerability value (0-2)	Asset value (0-4)	Risk value (0-8)
5	Loss of network connection	There's no mechanism for ensuring offline operation	Unattended payment terminal system	1	1	4	6
			Kiosk application	1	2	3	6
		There's no offline alerting mechanism implemented	System monitoring software	1	1	1	3
6	Misuse of allocated system privilege set	There is no clear separation between the sets of privileges available for POS staff and technical maintenance specialists	Operating system	1	1	4	6
		There are unnecessary privileges available in the administrator menu for the POS staff	Operating system	1	1	4	6
			Kiosk application	1	0	3	4
7	PIN theft	Plastic keys are used at the PIN pad for the installations in Finland allowing to read thermal signature	Unattended payment terminal system	1	0	4	5

No	Threat	Vulnerability	Assets affected	Threat value (0-2)	Vulnerability value (0-2)	Asset value (0-4)	Risk value (0-8)
8	Data theft or altering	There's no software or policy disallowing external devices included with the setup (vulnerability related to the internal staff)	Locally stored logs	2	1	1	4
			Promotion codes	2	1	1	4
			External API access credentials	2	1	4	7
		Windows keyboard exposed in the normal operation of the self-service kiosk software allows to view the clipboard history and get access to previously used inputs for Estonian installations	Locally stored logs	2	2	1	5
			Promotion codes	2	2	1	5
			External API access credentials	2	2	4	8
9	Exposure of the access credentials by employees having access to them	Employees having access to the sensitive information may not be fully educated on how to treat it securely	System access QR codes	1	1	3	5
			Operating system	1	1	4	6
			POS administrator	1	1	3	5

No	Threat	Vulnerability	Assets affected	Threat value (0-2)	Vulnerability value (0-2)	Asset value (0-4)	Risk value (0-8)
			POS staff	1	1	2	4
10	Unauthorized installation of software	Windows Developer Mode is enabled on the devices allowing non-certified applications to be executed for installations in Estonia and Finland	Operating system	1	0	4	5
		Microsoft Defender SmartScreen is disabled	Operating system	1	0	4	5
		Windows User Access Control (UAC) is disabled	Operating system	1	0	4	5
11	Zero-day attacks	Automatic updates for Windows 10 are disabled, and all the updates including security ones are performed manually during the closed hours	Operating system	1	1	4	6
		There is no framework update policy implemented	Kiosk application	1	1	3	5
			Print server application	1	1	3	5
			Hardware services	1	1	3	5

No	Threat	Vulnerability	Assets affected	Threat value (0-2)	Vulnerability value (0-2)	Asset value (0-4)	Risk value (0-8)
12	Unauthorized modification of system settings	Windows User Access Control (UAC) is disabled	Operating system	2	1	4	7

Appendix 3 - Risk calculation for the new generation of self-service software

No	Threat	Vulnerability	Assets affected	Threat value (0-2)	Vulnerability value (0-2)	Asset value (0-4)	Risk value (0-8)
1	Cross-site scripting attack, Code injection attack, OS command injection attack	Electron framework is powered by Node.js which allows it to execute system affecting commands	Operating system	2	1	4	7
			Web-based UI application	2	2	2	6
		Web-based applications allow referencing external untrusted resources	Operating system	2	0	4	6
			Web-based UI application	2	2	2	6
		Electron framework allows backdoor injection via altering electron.asar file without changing the end application signature	Operating system	2	2	4	8
			Web-based UI application	2	0	2	4
2	SQL injection attack	User inputs provided in UI application are not validated	Operating system	1	0	4	5
			Web-based UI application	1	2	2	5
			SQLite database	2	0	1	3
		User inputs provided via UI application are not sanitized	Operating system	1	0	4	5
			Web-based UI application	1	2	2	5
			SQLite database	2	1	1	4