UNIVERSITY OF VAASA

SCHOOL OF TECHNOLOGY AND INNOVATIONS

AUTOMATION AND INFORMATION TECHNOLOGY

Lassi Korhonen

SME ORIENTED INFORMATION SECURITY LEVEL MEASUREMENT INDICATORS

Master's thesis in Technology for the degree of Master of Science in Technology submitted for inspection

Vaasa 24.4.2020

Supervisor	Prof. Jouni Lampinen
Instructor	M.Sc. (Tech) Lauri Vehviläinen

PREFACE

I would like to thank my supervisor Professor Jouni Lampinen and instructor Lauri Vehviläinen for assisting me through the whole study. They gave me valuable comments and improvement proposals for the content of the thesis.

I would also like to thank my family and my girlfriend for supporting and pushing me throughout the whole study.

Vaasa, 19.4.2020

Lassi Korhonen

TABLE OF CONTENTS

PREFACE 2

TABLE OF CONTENTS 3

ABBREVIATIONS 7

ABSTRACT 8

TIIVISTELMÄ 9

1 INTRODUCTION 10

1.1 Background 11

1.2 Objective 11

1.3 Scope 11

2 SECURITY OF SYSTEMS 13

2.1 Definition 13

2.1.1 Information Security 13

2.1.2 Cyber security 14

2.2 CIA triad 15

2.2.1 Confidentiality 15

2.2.2 Integrity 16

2.2.3 Availability 16

2.2.4 Additions to the triad (Parkerian hexad) 16

2.3 Examples of different attacks types and threats 18

2.3.1 Injection attacks 18

2.3.2 Cross-site scripting (XSS) 19

2.3.3 Denial of Service (DoS) 20
2.4 Basic protection mechanisms 20
2.4.1 Updates 20
2.4.2 Encryption 21
2.4.3 Backups and recovery plan 21
2.4.4 Access controls 22
2.4.5 Continuous audits and reviews 22

3 SECURITY FRAMEWORKS 23
3.1 Definition 23
3.2 NIST cyber security framework 23
3.2.1 Structure 24
3.2.2 Usage 27
3.3 CIS critical security controls 28
3.3.1 Structure 29
3.3.2 Usage 31
3.4 Challenges 32
3.4.1 Lack of automation 32
3.4.2 Multilayered structure and implementation 33
3.4.3 Time consumption 33

4 DESIGN 35
4.1 Research plan 35
4.2 Requirements for the new security evaluation method 36
4.3 Input parameters 38
4.4 Calculation proposals 40
4.4.1 Base coefficients 40

4.4.2 Issue frequency based (Proposal 1) 41
4.4.3 Issue criticality based (Proposal 2) 41
4.4.4 Combination of issue amounts and criticality (Proposal 3) 42
4.5 Factor in business criticality and audit state in calculations 43
4.5.1 Per system 43
4.5.2 Per environment 43
4.6 Output 44
4.7 Security ratings 45

5 IMPLEMENTATION 46
5.1 Script implemented for testing 46
5.1.1 Python 46
5.1.2 Dradis API 47
5.1.3 Input 47
5.1.4 Output 48
5.1.5 Flowchart 48
5.2 Test case structures 50

6 RESULTS AND ANALYSIS 52
6.1 Test case 1 52
6.1.1 Proposal 1 52
6.1.2 Proposal 2 54
6.1.3 Proposal 3 56
6.2 Test case 2 58
6.2.1 Proposal 1 58
6.2.2 Proposal 2 60
6.2.3 Proposal 3 62

6.3 Test case 3 64

6.3.1 Proposal 1 64

6.3.2 Proposal 2 66

6.3.3 Proposal 3 68

6.4 Comparison and discussion between proposals 70

6.4.1 With base coefficients 71

6.4.2 Base versus modified coefficients 72

6.4.3 Modified coefficients versus weighted average 73

6.4.4 Criticism and comparison to available frameworks 74

7 CONCLUSIONS 76

REFERENCES 78

APPENDIX A 84

ABBREVIATIONS

API	Application Programming Interface
CA	Certificate Authority
CIS	Center for Internet Security
CVSS	Common Vulnerability Scoring System
DDoS	Distributed Denial of Service
DoS	Denial of Service
FIRST	Forum of Incident Response Security Teams
ICS	Industrial Control System
IoT	Internet of Things
IT	Information Technology
NIST	National Institute of Standards and Technology
OWASP	Open Web Application Security Project
SME	Small and Medium-sized Enterprises
SQL	Structured Query Language
XML	Extensible Markup Language
XSS	Cross-Site Scripting

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Lassi Korhonen
Topic of the Thesis: SME oriented information security level measurement indicators
Supervisor: Professor Jouni Lampinen
Instructor: M.Sc (Tech) Lauri Vehviläinen
Degree: Master of Science in Technology
Major of Subject: Automation and information technology
Year of Entering the University: 2015
Year of Completing the Thesis: 2020 **Pages:** 86

ABSTRACT

Information and cyber security activities has become a major part of companies processes because of the increasing number of devices and systems that are online and connected to internet. However, information systems should be under cyclic monitoring which implementation can be a demanding task for a small and medium sized company for example because of the lack of available resources. This study aims to develop a security evaluation method which can be easily integrated into small or medium sized companies' processes because of its straightforward and simple approach. Additionally, this study aims to develop a script which can be used for the testing purposes of the new security evaluation method.

The study was conducted by first gathering the requirements for the evaluation criteria which were then used to design the new security evaluation method. The inputs for the evaluation process were based on vulnerability severity and amount. The mentioned input attributes were also used to develop proposals for the calculation process. As an output the security evaluation method provided a single integer which described the target systems information security level. At the end of the study, the created security evaluation method was tested against three test cases which were used to evaluate the effect of different proposals and audit results for the output. A script was also developed for the testing purposes.

The results and analyses indicated that the developed security evaluation method was easier to integrate into small businesses processes because of its simplicity and straightforward approach. In addition, the issue frequency-based proposal which considered the business criticality and weighted average proved to be most suitable approach for the security state evaluation. However, the results also indicated that the weights used in the calculations should be raised in order to achieve a more descriptive picture of the state. The further development suggestions proposed that the created security evaluation method could be developed towards a more system-based approach instead of the case-based approach. In addition, the evaluation method could be integrated as a part of a service which calculates and evaluates the state of information security in the target system based on the input.

KEYWORDS: Information security measurement, Information security evaluation, Security frameworks

VAASAN YLIOPISTO
Tekniikan ja Innovaatiojohtamisen yksikkö
Tekijä: Lassi Korhonen
Tutkielman nimi: Kohdejärjestelmän tietoturvallisuuden mittaaminen pienissä ja keskisuurissa yrityksissä
Valvojan nimi: Professori Jouni Lampinen
Ohjaajan nimi: DI Lauri Vehviläinen
Tutkinto: Diplomi-insinööri
Oppiaine: Automaatio ja tietotekniikka
Opintojen aloitusvuosi: 2015
Tutkielman valmistumisvuosi: 2020 **Sivumäärä:** 86

TIIVISTELMÄ

Eri järjestelmien tietoturvasta on tullut iso osa organisaatioiden prosesseja internettiin kytkettyjen laitteiden ja järjestelmien määrän kasvun takia. Tietojärjestelmien tulisi kuitenkin olla säännöllisen valvonnan alla ja tämän toteuttaminen saattaa olla vaikeaa pienille ja keskisuurille yrityksille, esimerkiksi resurssien puutteen vuoksi. Tämän tutkimuksen tavoitteena on kehittää uusi yksinkertainen järjestelmän tietoturvan arviointimenetelmä, joka on helppo integroida osaksi pienten ja keskisuurten yritysten prosesseja sen yksinkertaisen lähestymistavan ansiosta. Lisäksi tavoitteena on kehittää skripti, jota voidaan käyttää uuden laskentatavan testaamiseen.

Tutkimus aloitettiin keräämällä vaatimukset kehitettävälle järjestelmän tietoturvan arviointimenetelmälle, joita käytettiin uuden mittaustavan suunnittelussa. Uuden mittaustavan syötteenä käytettiin auditointien tuloksia sekä haavoittuvuuksien määrää ja niiden vakavuutta. Mainituista attribuuteista kehitettiin myös laskentaehdotukset. Ulostulona laskentatavat tuottivat luvun, joka kuvaa kohdejärjestelmän sen hetkistä tietoturvallisuuden tilaa. Tutkimuksen lopuksi luotua arviointimenetelmää ja sen tuottamia tuloksia testattiin kolmessa eri testitapauksessa, joissa verrattiin eri laskentatapojen vaikutusta kohdejärjestelmän tietoturvan tason laskentaan. Testejä varten kehitettiin skripti, joka suoritti laskennan.

Tulokset ja niiden analysointi indikoivat, että kehitetty arviointimenetelmä on helpompi integroida osaksi pienen yrityksen prosesseja sen yksinkertaisuuden ja suoraviivaisuuden takia. Lisäksi parhaimmaksi tietoturvan laskentatavaksi tutkimuksessa olevista vaihtoehdoista osoittautua havaintojen määrään pohjautuva laskutapa, joka huomioi järjestelmän bisneskriittisyyden ja laskee koko ympäristön tietoturvan tason käyttäen painotettua keskiarvoa. Lisäksi tulokset kuitenkin osoittivat, että laskentaan käytettäviä painoja tulisi nostaa kuvaavamman tuloksen saamiseksi. Jatkokehitysideat sisälsivät ehdotuksen arviointimenetelmän jatkokehityksestä järjestelmäkohtaisempaan suuntaan, jossa tutkittaisiin järjestelmälle suoritettuja toimenpiteitä yksittäisten auditointien sijasta. Lisäksi kehitetty arviointimenetelmä voitaisiin integroida tulevaisuudessa osaksi palvelua, joka laskee syötteen perusteella kohdejärjestelmälle tietoturvallisuuden tason.

AVAINSANAT: Tietoturvan mittaus, Tietoturvan tilan arviointi, Tietoturvaviitekehys

1 INTRODUCTION

Nowadays different parties are connecting more and more systems and devices online due to the advances of the internet (Katole, Sherekar & Thakare 2018: 1). These systems can also contain sensitive information which is one of the most crucial resources. New technologies are also developed and together they are exposing the systems and applications used in them into variety of different threats such as information leaks and virus infections. (Golyash, Sachenko & Rippa 2011.) In addition, if the system is for example a public service which gathers sensitive data of its users, the data must be protected at rest and in transit with different mitigation processes such as access controls and encryption. Therefore, information and cyber security has become a major part of companies' processes. (Von Solms 2001: 215.)

In order to protect their information systems and data stored in them, companies can evaluate target systems state in terms of information security by executing audits to the target systems. These audits can be executed by utilizing different frameworks which are offered by different authorities. The frameworks can be used as the guideline when the target systems information security is evaluated (NIST 2018, a: 1). However, in some cases the frameworks can be challenging and time consuming to integrate into a small and medium sized company's processes because of their layered structure and size. Therefore, a new security evaluation method could be developed to ease the evaluation process.

This study will offer a solution for a smaller company to evaluate their information systems state in terms of information security. The thesis will cover basics of security, the already available frameworks, the design of the new information security evaluation method and script. The thesis will also introduce three different calculation proposals which can be used to evaluate the security of the target system. These proposals will be then tested against three different test cases. The results of the test cases will be then used to evaluate and analyze the created security evaluation method, the functioning of the script and to analyze the different calculation proposals together. At the end of the thesis, conclusions will be presented with further recommendations on how to utilize or develop the security evaluation method in the future.

1.1 Background

This study is done as an assignment to a private company. The topic came from the technical unit of the company because there is a need for automating and simplifying the processes in evaluating the target systems security state. In addition, the developed security evaluation method could be used for continuous security measurement of the target information system because of its simplicity and integration possibilities.

1.2 Objective

This study aims to develop and offer a solution for a smaller company to evaluate their information systems state in terms of information security. As a result, a new evaluation method will be designed, developed and proposed. The aim of the study is also to develop a script which can be used for the testing purposes of the new security evaluation method. In addition, this thesis will provide or offer ideas on the integration possibilities of the developed security evaluation method into company's processes in the future.

1.3 Scope

The scope of the thesis is primarily limited to the design of the new security evaluation method and to the implementation of the script. In addition, the result will only be compared to the two most utilized or common available frameworks which are the NIST (National Institute of Standards and Technology) cyber security framework and CIS (Center for Information Security) Critical Security Controls. Other possible security frameworks are not taken into account in the comparison.

The new security evaluation method is primarily developed considering the needs and requirements set by the company who ordered the thesis. Thus, the security evaluation method can mainly be utilized internally by the company. Additionally, the script which will be developed in the thesis is mainly intended for testing purposes. However, the script

will be designed and developed so that it can also be used in a production environment with minor changes. The further implementation or integration of the result is not included in this thesis and it will be left to internal development.

2 SECURITY OF SYSTEMS

Information and cyber security of different systems are a crucial part of companies processes due to increasing reliability on IT (Information Technology) and increasing complexity of different infrastructures (Bowen, Hash & Wilson 2006: 2). This chapter will cover basics of information and cyber security including the definition, the CIA (Confidentiality, Integrity, Availability) triad and different attack types and mitigation mechanisms.

2.1 Definition

Information and cyber security are often considered and misunderstood as the same thing. Although they both concern the security and protection of different systems, these terms are not equal and there is a clear separation between them (Fasulo 2019). Both terms are presented in the following chapters.

2.1.1 Information Security

Information security can be defined as the protection of information from unauthorized operations such as access, use and disclosure. The goal of information security related operations is to ensure and implement confidentiality, integrity and availability for the information in transit and at rest (Nieles, Dempsey & Yan Pillitteri 2017: 7). Abruptly, this means that the information in transit or at rest should not be altered in any way nor it should not readable by an unauthorized party. In addition, the objective of information security is to ensure that business related processes can remain continuous and damage impact to different systems are limited (Von Solms & van Niekerk 2013). For example, the National Institute of Standards and Technology (NIST) defines information security in the following way:

"The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability" (Paulsen & Byers 2019: 94).

2.1.2 Cyber security

As Information security is oriented towards all aspects of the information of data in terms of confidentiality, integrity and availability, cyber security considers almost everything that can be reached via cyberspace. Cyberspace can be defined as an operational environment which is formed from different digital information systems. This means that in practice cyber security considers the protection of different appliances and for example the security of critical computer infrastructures. (Von Solms & Niekerk 2013.) For example, the Kaspersky lab defines the cyber security in the following way:

"Cyber-security is the practice of defending computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks. It's also known as information technology security or electronic information security." (Kaspersky labs 2019.)

Cyber security can also be divided into six different domains or layers which are network, application, operational and information security and disaster recovery and end-user education. The network security layer attempts to secure the network from attackers and from intruders with different operations such as firewall policies. The application security domain attempts to protect the applications or software's that are running on different devices. The operational security includes the processes that are related to data handling. In addition, operational security takes into account different privilege levels and access controls. The disaster recovery (Incident response) examines how for example an organization reacts to a data breach or some other loss of data or operations. The domain also includes the policies for the restoring activities after the incident. End-user education examines the social side of the cyber security by taking into account the people that are interacting with for example some information system. By educating people on different threats such as attack types that can be executed for instance with email, the company can save itself from cyber security incidents. (Kaspersky labs 2019.)

2.2 CIA triad

The CIA triad forms the base for information security in the IT field. The triad concerns the confidentiality, integrity and availability of the target system or the data. These concepts can be viewed as the objectives regarding the IT field. However, the definition for the triad can differ based on the assets that they are targeting. (Oscarson: 4.) In this chapter, the triad is reviewed based on information assets. The basic CIA triad can be described for example with the figure below. (Figure 1.)

Figure 1. The CIA triad (Ko, Osei-Bryson & Dorantes 2006: 8)

2.2.1 Confidentiality

Confidentiality means that the information assets or the data in them is not disclosed by an unauthorized party including different individuals, entities and processes (Oscarson: 4). In other words, this means that the data which is in transit or stored for example into a database by some web application, cannot be read by an attacker which exploits some vulnerability that is present in the system or in the application itself. This security property indicates that if your data or information is not confidential it is considered as insecure (Pender-Bey 2012: 8).

2.2.2 Integrity

Integrity refers to the manipulation or modification of the data in transit or at rest in an unwanted way (Oscarson: 4). By achieving full integrity of the information asset, an unauthorized party should not be able to modify the data that is present in the application or in the system. In order to maintain integrity, there needs to be processes that will prevent undesirable actions or changes to the data. This can be achieved for example by using authentication mechanisms which prevent undesired changes to the data. In addition to the undesirable action prevention, the data should be easily recoverable. (Pender-Bey 2012: 12.)

2.2.3 Availability

Availability refers to the authorized access of the data. When full availability is achieved, the information asset should always be available when needed for the authorized party. (Oscarson: 4.) If the availability is affected for example by an attacker and users are not able to access an information asset at will, the total availability of the asset is lost. Any unauthorized accesses should be denied, and the system should also be resistant against different Denial of Service (DoS) attacks.

2.2.4 Additions to the triad (Parkerian hexad)

In 2002, information security consultant, Donn B. Parker introduced a more complex and comprehensive variation of the CIA triad known as the Parkerian hexad. The Parkerian hexad adds three more components or elements to the CIA triad which are the authenticity, utility and possession. The Parkerian hexad is presented in the figure below. (Figure 2.)

Figure 2. Parkerian hexad (Pender-Bey 2012: 7)

Authenticity adds the proof of identity to the triad. The main objective of the authenticity is to ensure that the message or transaction is from the source that it states to be. This property is highly critical when considering high security systems such as online bank services. The authenticity of a transaction can for example be verified with digital certificates which are issued by predefined and trusted Certificate Authorities (CA). The digital certificates are used to proof the identity of the service provider. (Pender-Bey 2012: 14.)

Utility property is used to measure the usefulness of the data. For example, if the data in transit is encrypted and the third party which receives the message does not know the key or the password to decrypt the message, the message comes useless to the receiver because it is in an unreadable state. (Pender-Bey 2012: 17-18.)

Possession property defines another component to the triad which is related to the possession of the data. The property claims that any confidential information or data can be controlled or be possessed by an unauthorized party without risking the confidentiality. However, in addition if the device which has the confidential data stored into a hard drive is stolen it offers the unauthorized party an opportunity to retrieve the data. This can be mitigated for example by enabling disk encryption. (Pender-Bey 2012: 11-12.)

2.3 Examples of different attacks types and threats

Attackers or malicious parties can attack information systems with various of different techniques. By leveraging for example different vulnerabilities which are present in the service, the attacker can cause serious damage to the confidentiality, integrity and availability of the target service. This chapter will only cover some of the most recognizable attack types and vectors which are injection attacks, cross-site scripting (XSS) and Denial of Service (DoS). The attack types are mentioned in an order which is compatible to CIA.

2.3.1 Injection attacks

Injection attacks are one of the most common attack types that are facing different applications nowadays. For example, OWASP (Open Web Application Security Project) has ranked injection-based attacks in their publications as the most crucial flaw that is affecting different information systems. OWASP is an association which purpose is to support the secure development of web applications. Different injection type attacks are for example SQL (Structured Query Language) and XML (Extensible Markup Language) injections. (OWASP 2017, a.)

Typically, an application is vulnerable to injection attacks when the user-supplied data is not filtered or sanitized. Therefore, the attacker can inject or manipulate the backend queries and force the application to return data for which the party is not authorized to access. This can lead to data loss, corruption or information disclosure to unauthorized parties. The impact for the business is very dependent of the criticality of the data. (OWASP 2017, a.)

The prevention for these types of attacks is heavily related to the arbitrary user input validation and keeping the supplied data separate from backend commands. The input validation could for example be implemented using a whitelist which only accepts certain characters. In terms of for example SQL injection attacks, parameterized queries can be used for mitigation. (OWASP 2017, a.) Parameterized queries are a method in which the

SQL statement is first precompiled or defined and after that the parameters are passed to the query (Reetz 2017: 3.)

2.3.2 Cross-site scripting (XSS)

Cross-site scripting (XSS) attacks are also one of the most recognizable and well known attack types in modern web applications (Pranathi, Kranthi, Srisaila & Madhavilatha 2018: 1). In a cross-site scripting attack, the attacker is able to inject malicious arbitrary JavaScript into the application which will then be executed by the browser with other users' privileges (CIS 2017, a: 1). An XSS flaw can be used to hijack users' sessions and steal other sensitive information stored by the browser (OWASP 2017, b).

There are three types of XSS which are reflected, stored and DOM (Document Object Model). In reflected XSS the user supplied script is reflected to the server's response without any sanitation or filtering. Therefore, when the application is rendered by the browser the malicious script will run. In stored XSS, the malicious script is saved for example into applications database without sanitation. Therefore, if client and server-side sanitations or filtering are not implemented, the payload gets executed whenever the application accesses the database and reflects the payload to the response. In DOM based XSS, the malicious script modifies the DOM environment so that the original client-side code is executed in an unexpected way which results into client-side code execution. (OWASP 2017, b.)

The XSS flaws can be mitigated for example by using input sanitation and filtering. In addition, different frameworks can be used in the development process to automatically escape the malicious characters from the user input and therefore prevent malicious script execution. (OWASP 2017, b.)

2.3.3 Denial of Service (DoS)

Denial of Service attacks are one of the most common attack types in which the attacker interrupts a services availability to legitimate users by consuming its resources. The number of Denial of Service attacks are also increasing because the attacks sometimes does not require much technical knowledge and automated tools are available for the execution. On a general level, there are two types of Denial of Service attacks which are network and application level attacks. In a network level attack, the attacker consumes the network bandwidth and computing resources of the target. In an application level attack, the attacker abuses and exploits a logical flaw that leads into increasing consumption of system resources. (Soliman & Azer 2018.)

Denial of Service attacks can also be distributed. These are called DDoS (Distributed Denial of Service) attacks in which the attacker uses multiple machines to attack into a certain target. (Gupta, N & Jain & Saini & Gupta, V. 2016.)

2.4 Basic protection mechanisms

Although, the malicious parties have many ways and techniques to exploit and abuse the vulnerabilities and misconfigurations in a system, the attacks impact can be mitigated by following certain principles. In this chapter, some mitigation techniques are presented including regular updates, encryption, backups, access controls and continuous audits.

2.4.1 Updates

By applying regular updates to the software's running in the information systems, a lot of different attacks can be mitigated. The usage of components with known vulnerabilities is one of the most common issues present in the systems according to OWASP (OWASP 2017, c). By using old components in the systems, the attackers can leverage and exploit already found vulnerabilities which are present in the outdated component and for exam-

ple get unauthorized access to the system. Although the dangers of using outdated components are known and the vulnerabilities could be easily fixed by just applying regular updates, many organizations are not implementing regular updates into their systems. (National Cyber Security Center 2018.)

2.4.2 Encryption

Encryption can be used to protect the confidentiality of the data. It is a mathematical operation which transform the plaintext data using a long enough key into a format which is not easily understood by people. After the transformation the output should not reveal or hint anything about the data that was transformed. In order to read the data again, the data should be decrypted using a key. In order to prevent the attackers from decrypting the data, the key must be secret and should be shared only with the involving parties. (Stine & Dang: 1-2.) The basic principle of encryption is figured below. (Figure 3.)

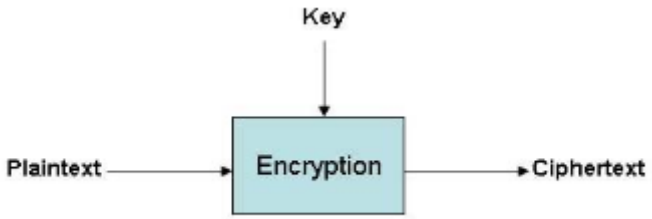

Figure 3. Encryption described in a general level (Stine & Dang: 2)

2.4.3 Backups and recovery plan

Backups are copies of files and programs which are used to recover from unspecified states. They can be used as a part of a recovery plan. Recovery plan is a set of procedures which main objective is to achieve full restoration after for example a data breach or malware infection. (Bartock, Cichonski, Souppaya, Smith, Witte & Scarfone 2016: 1.) According to the National Cyber Security Center (2018: 43) many organizations do not have a sufficient recovery and backup restoration plans.

2.4.4 Access controls

In order to prevent for example unauthorized operations, proper access controls must be implemented to the system. The main purpose for access controls is to limit the operations that a certain application or user can perform or execute. (Mudarri & Al-Rabeei 2015: 1.) Without proper access controls unauthorized users could read and modify data. Therefore, determining an access policy is crucial in terms of systems security.

2.4.5 Continuous audits and reviews

Information security audits and reviews can be used as tool to monitor the security level of a certain system. In a security audit some party examines the target infrastructure or system critically and reports the findings to the owner of the target. These findings can for example be vulnerabilities that are present in the service or policies that are not met. By executing these audits continuously and on a certain interval the target systems information security level can be monitored and enhanced in timely basis. It also offers the change to fix the reported security flaws before the application is taken into production.

3 SECURITY FRAMEWORKS

Different security frameworks can be used as a tool for security audits and reviews. The main purpose for the frameworks is to offer a solid base for determining the state of security in the target system. This chapter will define and examine the structure of two different security frameworks which are the NIST cyber security framework and CIS critical security controls. In addition to the definitions, the usage and structures of the frameworks are presented.

3.1 Definition

In general framework refers to a conceptual structure which can be utilized as a guideline for creating different concepts. As an output the framework usually extends the composition into something functional and useful. (Rouse 2015.) Hence, a security framework can serve as a model which strengthens the security of the target infrastructures (NIST 2018, a).

3.2 NIST cyber security framework

The National Institute of Standards and Technology has developed a framework which main purpose is to enhance the security risk management in target systems and infrastructures. The framework will provide an organizing structure for different approaches in terms of security by combining working standards, guidelines and practices productively and flexibly together. It is designed to function regardless of their focus in security in the target organization. Thus, the framework can for example be applied in IT (Information Technology), industrial control systems (ICS) and Internet of Things (IoT). Although the framework can be applied to multiple different use-cases, it cannot be used as an inclusive approach because the technologies that are affecting target systems are always evolving and thus creating new different attack vectors and risks. (NIST 2018, a.)

The framework will use the standards and guidelines to determine current security posture and state of the target. In addition, it attempts to identify and prioritize the parts that can be used in continuous improvement processes and investigate the progress towards the target security state of the system. It also offers new angles on how to communicate different security risks internally and externally. (NIST 2018, a.)

By now the National Institute of Standards and Technology (NIST) has released two different versions of the framework which are the versions 1.0 and 1.1 (NIST 2018, a). The most recent one (version 1.1) will be presented and gone through in this chapter.

3.2.1 Structure

The framework can be implemented by following a risk-based approach. It consists of three different components which are the core, implementation tiers and profile. (NIST 2018, a.)

The core component contains the activities, outcomes and suitable references that are universally separated between different critical infrastructures. It also addresses the communication of security activities across the whole target organization. These requirements have been transformed into five different functions which are called identify, protect, detect, respond and recover. (NIST 2018, a.)

The identify function is used to refine the organizational consensus in understanding on how to manage different risks, systems, people and data. It helps the organization to better identify and understand the business context and their resources in terms of security risks. (NIST 2018, a.)

The protect function is used for the detection of different access controls or safeguards concerning the organizations resources. It endorses to limit the impact of a cyber-security event. (NIST 2018, a.)

The detection function concerns monitoring activities which are present in the target systems. It supports the development and implementation of proper cyber event and anomaly identification. (NIST 2018, a.)

The respond function states and supports the activities after a cyber-event or anomaly have been detected. These activities can be used to decrease the impact of the cyber security incident. (NIST 2018, a.)

The recover function supports the development of activities after a cyber-event has happened. It is used to help maintain plans and restoration capabilities of the affected system. (NIST 2018, a.)

The functions are also divided into categories and subcategories which purpose is to identify the hidden outcomes. Example of a category in the protect function could be "access control" which subcategory could be "data at rest is protected". These outcomes are then matched with different informative references which can be existing standards or guidelines concerning the subcategories. (NIST 2018, a.) A picture of the framework core is presented in the figure below. (Figure 4.)

Figure 4. The core structure of NIST cyber security framework. (NIST 2018, a)

Another component in the framework is the implementation tiers which offer angles on how a company could view the underlying risks and how the risks should be managed. They attempt to provide solutions and determine the extent to which security management is informed. In addition, it tries to determine how the risk management process is integrated into companies' activities. The tiers vary between tier 1 and tier 4 (partial, risk informed, repeatable and adaptive): tier 1 being the most concise where the organizations risk management processes are not planned and tier 4 being the most comprehensive in which the organization cyber security related activities are designed and continuously improved. The tiers are also divided into three components which are the risk management processes, integrated management program and external participation. (NIST 2018, a.) A figure of the tiers is presented below. (Figure 5.)

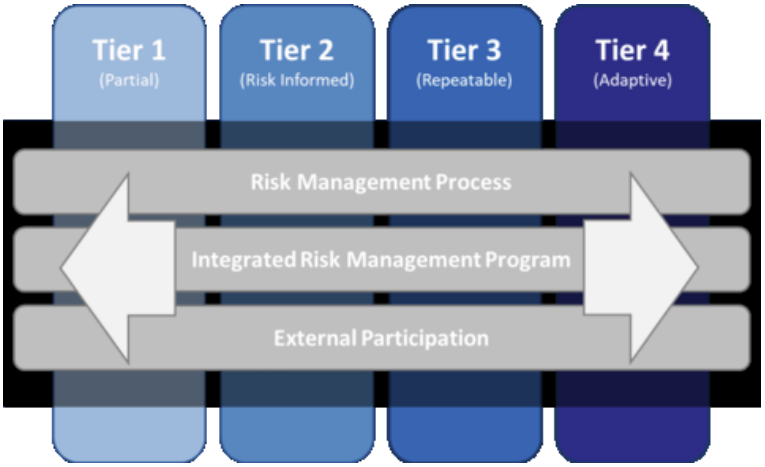

Figure 5. Implementation tiers (NIST 2018, b.)

After the target organization has investigated its internal and external processes concerning security, the framework can be used to create a profile which can be utilized as a roadmap for decreasing the possibility of cyber threats. The profile could include for example goals which should be met. Depending on the complexity of the target company, multiple profiles can be utilized and created (NIST 2018, a.)

3.2.2 Usage

In order to utilize the full potential of the framework, the National Institute of Standards and Technology suggests that a seven-step plan could be used in the implementation process. (NIST 2018, a.)

Prioritizing and scoping is the first step. This step includes the identification of the organizations business priorities. By starting the framework implementation process from this, the organization can make a more detailed and strategic decisions considering the security of present systems. (NIST 2018, a.)

Orient is the second step, which includes the identification of related systems, requirements and the overall risk approach. The organization can use consultation in the identification process. (NIST 2018, a.)

Creation of the current profile is the next step. This step includes the current security profile determination. This profile can be used later when the evolvement of security is examined. In addition, it will describe the current status of the security in the organization by determining the categories and subcategories which are currently reached. (NIST 2018, a.)

The risk assessment should then be conducted in which the organization evaluates and analyzes the target operational environment and determines the probability of a security incident. The organization can make the evaluation internally or use external sources such as consulting in order to gain a broader view of the threats. (NIST 2018, a.)

New target profile is created after the risk assessment step is done. In this step the organization should focus on the categories and subcategories of the framework and determine the desired outputs for them. (NIST 2018, a.)

Determine, prioritize and analyze gaps is the sixth step. In this step the organization uses the current profile created in the third step and determines, analyzes and prioritizes gaps compared to the new target profile. This comparing process will be beneficial in

determining the security gaps. This step also includes the designing process of the action plan which has the desired improvements in a prioritized order. (NIST 2018, a.)

Action plan implementation is the last step. In this step the organization identifies the gaps and deficiencies and adjust the current cyber security practices in order to fulfil the objectives determined in the target profile. (NIST 2018, a.)

The National Institute of Standards and Technology (NIST 2018, a) also addresses that the implementation of the framework could be continuous and executed in certain intervals in order to achieve improvement in the security aspect. Organization could also choose the steps which are crucial to them and monitor their progress (NIST 2018, a).

3.3 CIS critical security controls

CIS (Center for Internet Security) is a non-profit association which objective is to protect the private and public organizations from cyber threats. The association has developed multiple benchmarks and guidelines which can be used in securing IT systems and the data stored in them. (CIS 2019, b) One of these guidelines is called critical security controls which includes multiple different entities which can be used to improve the security of the target system.

The CIS critical security controls is a framework which contains 20 recommended actions for the mitigation of different cyber threats and malicious actions. The different components are gathered and analyzed from the most common attack vectors used and reported in the industry. By splitting the framework into 20 different components, it allows the organization to focus into a smaller segment of actions. This can enable the organization to achieve better results. (SANS 2020)

The critical security controls framework is under continuous development and the CIS association has published multiple versions of it. In this chapter, the newest version available during the written process of this thesis (version 7.1) will be evaluated and analyzed.

3.3.1 Structure

The framework consists of three different base entities which are the basic, foundational and organizational controls. These controls include subcategories or components which are each assessing a certain feature in the target environment. The structure of the framework is figured below. (Figure 6.)

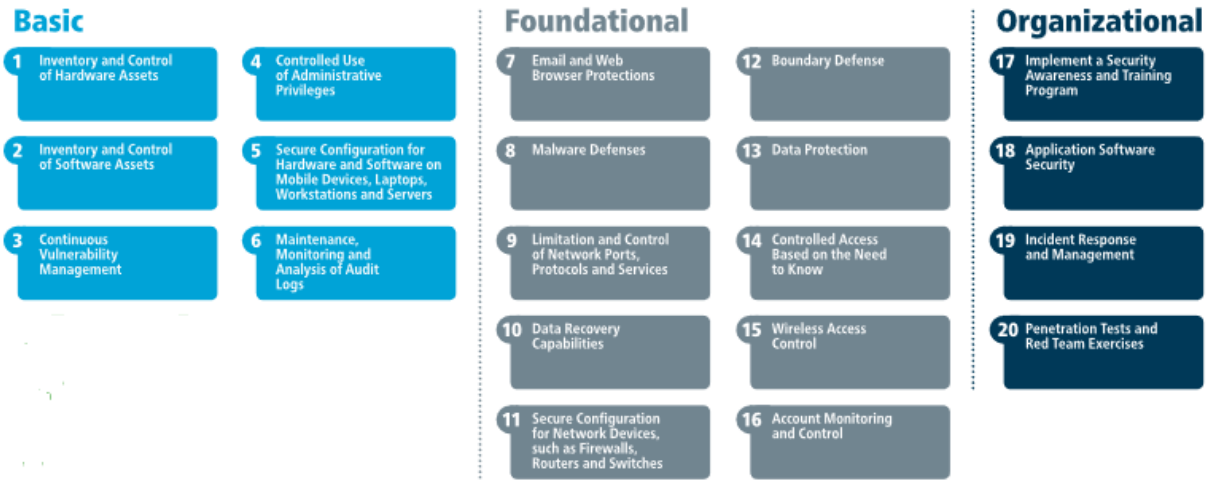

Figure 6. CIS cyber security controls components (CIS 2019, c)

The basic control includes trivial features or actions which should be considered whenever systems are deployed in a secure manner. The basic control includes the inventory and control of hardware and software assets, continuous vulnerability management, controlled use of administrative privileges, secure configuration and the monitoring and maintenance of audit logs. (CIS 2019, c.)

The inventory component addresses the access controls of the hardware and software's which are present in the target environment. It also specifies the identification of the unauthorized and unmanaged devices and software's. The continuous vulnerability management control indicates that the target environment should be continuously audited in order to identify and fix new threats or vulnerabilities that are present in the implementation. This can prevent the attackers from gaining access to the system and protect the system from different emerging threats. (CIS 2019, c.)

In order to enhance the security in the basic level, the systems need to be configured securely and also monitored and maintained continuously. These procedures can help the organization to detect, understand and recover from a cyber-attack. (CIS 2019, c.)

The foundational control extends the basic control by adding data protection fundamentals and access control implementations to the security of the system. This control includes the email and web browser protections, malware defenses, limitation and control of network ports, protocols and services, data recovery, secure configuration of network devices, boundary defenses, data protection, controlled and wireless access and account monitoring and control. (CIS 2019, c.)

The first three components in the control (email and web browser protections, malware defenses, limitation and control of network ports, protocols and services) are heavily associated with the mitigation and reduction of the attack surface. For example, the email and browser protection component considers the attacker's human behavior manipulation and using the email systems as an attack surface. In other words, the component attempts to ensure that only browsers and email clients which are fully supported and allowed internally in the organization are utilized and executed. (CIS 2019, c.)

The last components in the control (secure configuration, boundary defense, data protection, controlled access, wireless access and account monitoring and control) excluding the data recovery capabilities which is related to backup policies, are associated with access controls and the detection of malicious activities in the environment. For example, these components consider which ports should be closed and which applications and users should have rights to execute in the environment. (CIS 2019, c.)

In addition to basic and foundational controls, the organizational control adds the implementation of the incident response plan and continuous audits to the company's processes. This control includes the implementation of security awareness and training programs, application software security, incident response and management and penetration tests and red team exercises to the control. These components will cover the identification

of security gaps in the organization and the management of software life-cycle by executing security reviews. (CIS 2019, c.)

3.3.2 Usage

The objective of the CIS critical cyber security control framework is to offer a small, prioritized list of actions which can be used to improve the security state of the target environment. In addition, the framework offers new angles in terms of security which should be taken into account when the organizations security management is considered. However, CIS (2019, c) underlines that the framework is not applicable to every situation and the user should understand the target environment fully before implementing the framework into organizations processes.

The framework also adds different implementation groups for the prioritization of the different controls and components. These can be used as a methodology in the implementation process. The different implementation groups are shown in the figure below. (Figure 7)

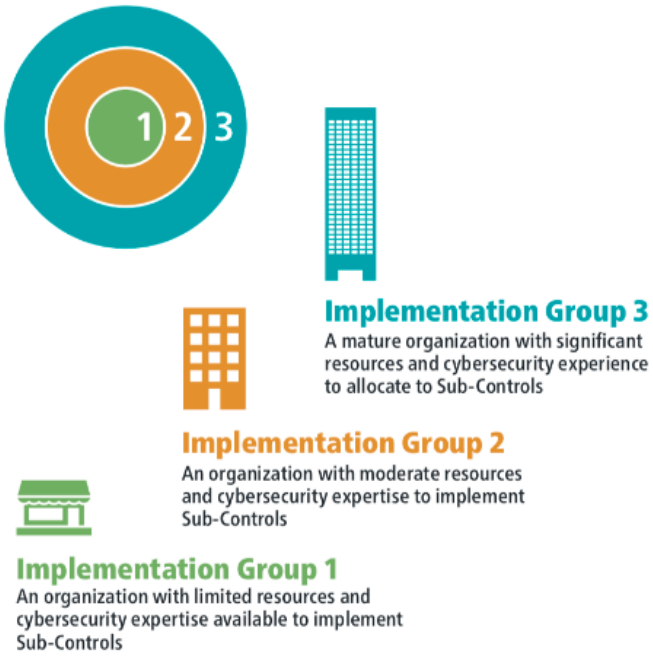

Figure 7. Implementation groups (CIS 2019, c.)

The implementation group 1 mainly considers small companies with approximately 10 employees. In addition, the controls included in the group 1 are essential to success in the security field and these should be implemented into every company's IT security program. These controls are for example continuous automated software updates and active logging. (CIS 2019, c.)

The implementation groups 2 and 3 mainly consider larger organizations which can have hundreds of employees, large amount of resources and complex IT systems. However, controls included in these groups can also be implemented into a smaller company's processes if necessary. (CIS 2019, c.)

3.4 Challenges

Although different security frameworks offer a great base and guideline to different organization in terms of information and cyber security, the implementation of the frameworks can have some challenges in some cases. These challenges are for example the lack of automation in the processes, multilayered structure and time consumption.

3.4.1 Lack of automation

In some cases, the implementation of the security frameworks can require a lot of manual work. For example, some of the different issues that are observed in the target environment are usually gathered and reported manually in these frameworks. In addition, the different functions and controls are executed manually. This manual work is targeted towards the organization internal or external resources depending on the party which is executing the audit or review. The audit or review can be executed by the organization's internal security section or by a third-party security company.

Additionally, because of the lack of automation, the frameworks can be difficult to implement into continuous monitoring activities. The frameworks offer a detailed overview

of the security state of the system in that specific time when the framework is implemented and used in the evaluation process. However, in order to achieve a detailed description of the security state continuously, the framework should be implemented manually in a certain interval. Because of the manual load, the implementation can be time consuming and difficult to a smaller company.

In addition to the issue reporting, the frameworks do not automatically evaluate or measure the information security state of the whole target system. This evaluation process of the security state is left for the organization itself. For a small organization this can be an overwhelming process when resources are considered.

3.4.2 Multilayered structure and implementation

The structure of the frameworks can be complex because they include a lot of different functions and controls which will act as the main categories. In addition to the main categories, subcategories are included to the main functions which splits the functions into smaller domains. Thus, implementing the framework into a smaller businesses processes can be difficult in some cases.

Additionally, because of that the implementation of the framework can be more challenging for a smaller company. Although in theory and in practice by splitting the functions into smaller domains, the framework will create a more comprehensive model of the security state of the target system thus reducing the risk of leaving security misconfigurations to the system.

3.4.3 Time consumption

Because of the multilayered structure and lack of automation, the implementation process of the framework can be time consuming. Hypothetically, limited resources in smaller businesses can also affect the time consumption by some extent. Additionally, in some SME oriented companies the resources can be bound to business development activities and the general know-how is focused around the core business activities. Therefore, if the

core business of the company is not focused around information and cyber security, the implementation of a framework can be a demanding task for the company. In these situations, consultation can be used. However, the consultation process can still demand a lot of time.

4 DESIGN

This segment of the thesis is going to present the design of the new security evaluation method. The topics presented and discussed in this chapter are constructed based on the interviews and to own observations and conclusions regarding the design of the new security evaluation method. First the requirements of the new evaluation method are presented thoroughly. These requirements are then used to design and implement the core of the evaluation method.

The core includes the inputs, calculations and output. The inputs have been gathered by considering different attributes of audit results. In addition, the calculations have been designed based on the audit results.

Finally, the output will contain the security scores of every system, the whole environment score and graphs which are based to the calculation outputs. The challenges of the already available frameworks have been taken into account in the design process.

4.1 Research plan

This study will be conducted by first gathering the requirements from the principal which ordered the study. The requirements will be gathered by using qualitative methods such as interviews and in addition by analyzing the already available frameworks.

After the requirements for the new security evaluation method have been gathered, they will be analyzed and used to design the new evaluation methods calculations and the script which will be developed for testing purposes. The calculations will include three proposals which functioning and descriptive properties will be analyzed.

Finally, after the new security evaluation method have been designed and the script have been developed the developed proposals will be run against three different test cases which are designed to describe the different characteristics of each calculation proposal.

4.2 Requirements for the new security evaluation method

This chapter covers the technical and conventional requirements for the new information security evaluation method. These requirements have been gathered from the principal who ordered the thesis and by examining the challenges of implementing a different framework. The requirements are not in order of importance

Ease of implementation is the first requirement for the new security evaluation method. The new information security evaluation method has to be easy to be implement into company's processes. In practice this means that the evaluation method should not be too complex and layered. In addition, the company or organization who decides to use the evaluation method should be able to recognize the problems in their systems easily. By designing the evaluation method as simple and straightforward as possible, the companies will also obtain benefits in terms of time consumption. The savings in time consumption can then translate into better results in terms of profitability and performance.

Automation is the second requirement for the new security evaluation method. The available frameworks lack the automation features and they can require a lot of manual work if the target system or environment is extensive and complex. The new security evaluation method should be designed so that it can be automated easily. Therefore, the architecture and calculations in the new security evaluation method should be straightforward and understandable. The calculations and the evaluation of the security state of the target environment should be done automatically. In addition, the monitoring of the security state of the target environment should be continuous. The automation will likely benefit the performance and profitability of the implementation of the new information security evaluation method.

Continuous monitoring is also a requirement. The new system security evaluation criteria should be feasible to implement as a part of company's continuous monitoring processes. Herewith it should calculate a security score for the system in a certain, feasible interval.

Maintainability is also considered as a requirement for the new security evaluation method. The script used to evaluate the systems security level should be developed so that it is easy to maintain and to develop further. Herewith the scripts architecture and design should be clear, comprehensible and it should support automated testing activities.

Unknown information systems which are not audited and are present in the environment should also be factored in the calculation processes. By factoring unknown systems in the calculations, the company which utilizes the developed system security evaluation criteria will get a broader understanding of the whole environment. Therefore, in addition, the company can for example plan, design and support its system security enhancement and development activities in the future.

Distinct categories should also be considered in the calculations. Security audits can be divided into different categories by factoring distinct characteristics and operations executed during the audit. For example, technical audits can contain manual web based penetration testing, scans and architectural reviews of the target environment, information security management assessments contain thorough reviews of the target organization activities considering also the business assets and evaluates whether certain security functions such as risk controls and incident response plans are addressed properly and are compliant with corporate policy (Ghazouni, Medromi, Boulafdour & Sayouti 2013: 1).

Audit type independence should also be considered in the calculations. The calculations executed in the security level evaluation process should not be independent of the audit type. In other words, this means that the calculations should be able to implement for any audit type or category.

Business criticality of different systems can have effect on the security evaluation process. Organization can have numerous distinct applications which will have different business criticality based on the missions they support (Paulsen, Boyens, Bartol & Winkler 2018). For example, a banking application has a raised business criticality rate because it processes and contains highly sensitive data of its users. On the other hand, a normal static website which does not handle any sensitive data has a normal business

criticality. Therefore, systems present in the target environment should be divided into distinct levels in order to achieve a more thorough understanding of the whole environment and the state of its applications. By dividing systems into business criticality levels, it helps the organization to plan and prioritize more business critical systems over other systems (Paulsen, Boyens, Bartol & Winkler 2018).

Scalability properties are important if the security evaluation method will be developed in the future. The new information security evaluation method should be developed so that it can be scaled to assess security in a more comprehensive level than in single system level. Thus, the calculations should be developed so that the security state will be calculated in both system and environment levels. The score of the environment can then be used to address the whole security state of the target company.

4.3 Input parameters

This chapter covers the distinct input parameters and attributes which are considered in the system security level evaluation process. The inputs will be presented in an input file. The parameters are described in a security audit level which means that every audit is presented in distinct rows in the input file and every entry must implement attributes described in this chapter as parameters excluding the systems which have not been audited yet but are present in the environment.

Timestamps will describe when the audit has been executed. The timestamps should be represented and appended in string format (Year-Month-Day) to the input file. Timestamps will be used for determining the elapsed time between the audits and for generating security scores monthly for every system.

Audit categories should also be considered as an input. Audits can be divided into multiple different categories based on audit characteristics and type. In this case, the division is made on a very general level. Thus, audits are divided into technical audits, information security management and scans.

CVSS scores (Common Vulnerability Scoring System) is used to determine the characteristics and severity of different vulnerabilities in the target software. The scoring system consists of three distinct metric groups which are base, temporal and environmental. The base metric describes the characteristics of single a vulnerability. For example, it evaluates the vulnerability by considering the loss of confidentiality, integrity and availability caused by the flaw.

The temporal metric considers the characteristics of the vulnerability that change over time. These characteristics include for example the availability of exploit code which is bound to time. The more time the malicious users have to develop exploits targeting a single vulnerability, more likely exploits will be available. The environmental metric considers the characteristics directed to single system. For example, it includes the business criticality of a system into the calculations. As a result, the calculations will generate a value between 0 and 10. The more severe the vulnerability is, the closer the end result is to 10. (Forum of Incident Response Security Teams (FIRST) 2019.)

CVSS scores are used to differentiate various vulnerabilities from each other and to divide vulnerabilities into critical, major and minor flaws and issues. The thresholds used in the division are partly taken from the CVSS scoring system (Forum of Incident Response Security Teams (FIRST) 2019). The minor and major ratings are the same but the high and critical ratings are merged into one group. The high and critical ratings are merged in order to achieve a simpler design. The thresholds used in the calculations are described in the table below. (Table 1.)

Table 1. Vulnerability thresholds

Rating	CVSS score
Low	0.1 - 3.9
Medium	4.0 - 6.9
Critical	7.0 - 10

Vulnerability frequencies per rating will be considered also as an input parameter. The ratings will be based on the low, medium and critical ratings (Table 1.). Vulnerabilities will be calculated and divided per audit to the input file.

Audit state evaluation should also factor in the unknown systems which have not been audited. This requirement will be described by this parameter. If the system is audited the value will be 1 and if it is not, the value will be 0.

Business criticality of the target system will consider the importance and criticality of the environment to the customer. This metric will have three levels which are high, moderate and low. The calculations will modify the weights based on these attributes.

4.4 Calculation proposals

The security state evaluation of a target system will be done using mathematical operations which consider different audit relative attributes such as the frequency of issues and the criticality of a single vulnerability. In this thesis three different calculation proposals will be presented and examined. These proposals include the issue frequency based, issue criticality based and the combination of both the issue frequency per category and criticality. The proposals have been constructed by considering the different results which the audit generates.

4.4.1 Base coefficients

Coefficients are used in the calculations to weight more severe vulnerabilities over minor rated vulnerabilities. This is based on the assumption which specifies that critical vulnerabilities will have a more significant effect on the whole security state of the system than minor vulnerabilities. As a result of that, the more the target application or system has more severe vulnerabilities the lower the final security score will be. The used coefficients are described in the below table. (Table 2.)

Table 2. Base coefficients

Rating	Coefficient
Low	0.5
Medium	1.0
Critical	1.5

4.4.2 Issue frequency based (Proposal 1)

The first calculation proposal includes the issue frequency per vulnerability category. These categories are low, medium and critical and the thresholds are the same as described in Table 1. All the vulnerabilities found in the specific audit are divided into those categories based on the thresholds and the amount is calculated by adding together all observations in specific category. The amount of vulnerabilities in every category is then multiplied by the coefficients (Table 2.) corresponding the category rating. This means that the amount of critical vulnerabilities is multiplied by 1.5, the amount of major vulnerabilities is multiplied by 1.0 and the amount of minor vulnerabilities is multiplied by 0.5. After that the results of every multiplication are added together and subtracted from 10. Thus, the final result will be in the range of zero to ten. If the final result after the subtraction is negative, the result is rounded up to zero. In addition, the security score will decrease by 0.3 monthly if the audit has been executed over a year ago. The calculation formula is presented below.

$$f(t) = \begin{cases} 10 - (1.5 * \sum criticals + 1.0 * \sum majors + 0.5 * \sum minors) & , 0 \leq t \leq 12 \\ 10 - (1.5 * \sum criticals + 1.0 * \sum majors + 0.5 * \sum minors) - 0.3 * (t - 12), t \geq 13, \end{cases} \quad (1)$$

where *t* is time in whole months (in whole numbers).

4.4.3 Issue criticality based (Proposal 2)

The issue criticality-based approach considers the CVSS scores calculated for every issue. The issue ratings and thresholds are the same as described in Table 1. The calculation process begins by factoring in all the issues in every category and dividing the CVSS

scores by 10. After the division, the modified scores will then be added together and multiplied by the coefficient which corresponds the addressed category rating. The results of every multiplication will then be added together and subtracted from 10. As in the first proposal, this proposal also includes the rounding up to zero if the end result is negative. Like in proposal one, the security score will decrease by 0.3 monthly if the audit has been executed over a year ago. The calculation formula is presented below.

$$f(t) = \begin{cases} 10 - (1.5 * \frac{\sum critical\ CVSS}{10} + 1.0 * \frac{\sum major\ CVSS}{10} + 0.5 * \frac{\sum minor\ CVSS}{10}) & ,0 \leq t \leq 12 \\ 10 - \left(1.5 * \frac{\sum critical\ CVSS}{10} + 1.0 * \frac{\sum major\ CVSS}{10} + 0.5 * \frac{\sum minor\ CVSS}{10}\right) - 0.3 * (t - 12),\ t \geq 13, \end{cases} \quad (2)$$

where *t* is time in whole months (in whole numbers).

4.4.4 Combination of issue amounts and criticality (Proposal 3)

The third proposal includes the combination of both the issue amount per category rating and the CVSS scores. The calculation process begins by dividing the CVSS scores by 10 and adding them together in every issue category. After that the result of the addition is multiplied by the issue amount in the category and then with the coefficient which corresponds the category rating. The result of every multiplication is then added together and subtracted from 10. Like in the first and second proposal, the result is rounded up to zero if the result is negative. Like in previous proposals, if the audit has been executed over a year ago, the security score will decrease by 0.3 monthly. The calculation formula is presented below.

$$f(t) = \begin{cases} 10 - (1.5 * \frac{\sum A}{10} * \sum D + 1.0 * \frac{\sum B}{10} * \sum E + 0.5 * \frac{\sum C}{10} * \sum F) & ,0 \leq t \leq 12 \\ 10 - (1.5 * \frac{\sum A}{10} * \sum D + 1.0 * \frac{\sum B}{10} * \sum E + 0.5 * \frac{\sum C}{10} * \sum F) - 0.3 * (t - 12), t \geq 13, \end{cases} \quad (3)$$

where *A* represents critical CVSS scores, *B* represents major CVSS scores, *C* represents minor CVSS scores, *D* represents critical vulnerabilities, *E* represents major vulnerabilities, *F* represents minor vulnerabilities and *t* is time in whole months (in whole numbers).

4.5 Factor in business criticality and audit state in calculations

One of the requirements for the new security evaluation process was that it should also consider the business criticality and audit state of a specific system in the calculation process. The business criticality and audit state consideration will be considered and examined per system and environment level. These alternatives will be integrated and tested together with the calculation proposals described earlier.

4.5.1 Per system

Per system level the calculation coefficients will be modified according to the business criticality of the system. In this case the business criticality levels are divided into three distinct categories which are high, moderate and normal. Described in a general level, the coefficient values will be raised if the business criticality of the system is higher. The business criticality levels, and the corresponding coefficients are described below. (Table 3.)

Table 3. Coefficients for different business criticality levels

Business criticality rating	Critical	Major	Low
High	1.7	1.2	0.7
Moderate	1.6	1.1	0.6
Normal	1.5	1.0	0.5

In addition, the security evaluation process will consider unknown systems which have not been audited yet. In system level the not audited entities security score will be flagged as 0 because without a properly executed audit against the system, the security state cannot be evaluated precisely.

4.5.2 Per environment

The business criticality per environment will be considered using weighted average. In general, weighted average refers to a function in which the whole vector of data points is

assigned with different weight and divided by the vector of weight data points (Bonfietti & Lombardi 2012). In this case, the weighted average is calculated weighing the more business critical systems over normal systems by multiplying the calculated security rating of the system with the corresponding weight value of the business criticality rating. The weights assigned to different system with distinct business criticality are described below. (Table 4.)

Table 4. Weights for business criticality levels.

Business criticality rating	Weights
High	2.0
Moderate	1.5
Normal	1.0

The not audited entities will affect the averages as a lowering factor only after system has been implemented to the environment because system cannot have an effect to the whole score if it does not exist. Therefore, the timestamps for these kind of audits should be set accordingly to the implementation date.

4.6 Output

As an output the new security evaluation process will generate security scores for system level examination and averages for environment level examination. In addition, graphs will be generated to support and easier the examination process.

Security scores will be calculated for every system in every category in monthly interval. This is done due to the requirement that every systems security score will decrease if the audit has been executed over a year ago. After a year the security score will decrease by 0.3 every month until a new audit is executed to the system. If the final score is negative, the result will be rounded to zero. In addition, the highest value of the score is ten.

The **averages** will be calculated monthly for every environment, audit type and group. The averages in these sections are calculated using arithmetic mean in which all the system scores in certain month are added together and divided by the frequency of systems in the data set. In addition, the weighted average which assigns weights into the security scores will also be calculated and examined in some test cases.

As an output, the security state evaluation will also generate **graphs** in order to create better and more wider understanding of the security state. The graphs will be generated for the whole environment, audit type and groups. The whole environment and audit group graphs contain only the monthly averages. Audit type graphs contain the monthly security scores of every case belonging to that audit type.

4.7 Security ratings

Every system will get a security score which will be used to describe the overall security state of the system. Additionally, every environment will get a security score which is calculated by taking the average between systems present in the target environment. In order to better evaluate the overall security state of the target system and environment, the security scores generated by the calculations should be mapped into ratings and states. The evaluation process will follow the below table in both system and environment levels. (Table 5.)

Table 5. Security ratings for environments.

Score	Rating/State
9-10	Excellent
7-9	Good
5-7	Moderate
3-5	Insufficient
1-3	Alarming
0-1	Critical

5 IMPLEMENTATION

This section will cover the implementation and design of the tests for every calculation proposal. The implementation phase of the tests will begin with the design of different test cases that will be used to test the different calculation proposals. The test cases will include tests that are designed to describe the effects of different calculation proposals to the security score. Additionally, the test cases should illustrate the effects of business criticality to the end result. After the test cases have been designed the implementation will continue with the design and development of the script which will be utilized for the testing activities. The script will also be designed and develop so that it can also be utilized in production with minor changes. The development will be done using python.

5.1 Script implemented for testing

Automation was one of the requirements for the new security evaluation process. This requirement will be fulfilled by designing and implementing a script which handles the calculation process and generates the desired output. In addition, by implementing a script, the ease of implementation and continuous monitoring requirements will be fulfilled. Indefinitely, the script will be developed mainly for the testing purposes of the designed security evaluation process in this study.

5.1.1 Python

Python is used to develop the script. It is a general-purpose programming language which can be utilized in variety of different software projects. However, Python is more often applied in scripting roles and it is often used because of its readability, productivity, portability and large collection of different libraries and integration possibilities. (Lutz 2009: 3-6) Thus, because of those features, it was also selected as the main programming language in this project.

5.1.2 Dradis API

Dradis API (Application Programming Interface) is used for testing purposes to generate arbitrary input data for the developed script. Dradis is a reporting and collaboration framework which is mainly utilized by information security teams (Security Roots Ltd 2020). The main reason for the utilization is that Dradis already contains the required data for the input file generation including the CVSS scores, titles and timestamps. However, the data queried from Dradis will be masked and obfuscated with arbitrary content so that it does not refer to original customer cases in any way. This will be done in order to ensure that real customer data is not used for testing purposes.

5.1.3 Input

The input file for the script will contain all the parameters and attributes which the new security state evaluation process will take as an input. These attributes are timestamp, audit category, CVSS scores or vulnerability rating frequency depending on the calculation proposal, audit state and business criticality of the target system. However, if the audit is a security review, the issue frequencies are inserted to the issue category columns instead of CVSS scores. In this case the format of the input file will be Microsoft Excel Workbook (XLSX). The data will be structured to the file in a manner in which every entry will implement all the attributes described earlier. A figure containing an example of the input file is presented below. (Figure 8.) The example contains only the first five entries.

	A	B	C	D	E	F	G	H	I
1	**Case**	**Type**	**Timestamp**	**Criticals/High**	**Majors/Medium**	**Minors/Low**	**Group**	**Audited**	**Criticality**
2	Case4	audit	2010-01-01				Internal Services	1	normal
3	Case3	audit	2010-01-01				Internal Services	1	moderate
4	Case3	audit	2010-02-01	7.7	5.4,4.1,4.3,5.0	3.6,3.7,3.3	-	1	normal
5	Case2	audit	2010-03-01				Continuous development	1	normal
6	Case4	scan	2010-03-01	7.5,9.1		2.8	Internal Services	1	high

Figure 8. A sample of the input file

5.1.4 Output

As an output the script will generate an excel workbook file which contains the security scores for every system or entity. Every system is also divided into different sheets by audit type. This means that for example technical audits, information security management audits and scans are divided accordingly into distinct sheets. Additionally, the output file will present averages calculated for every audit group and type. The audit group averages will be presented in the same sheet as the system for which the audit with the specified group has been executed. Averages for the whole environment will be presented in a dedicated different sheet. Finally, the security scores for every system will be generated monthly until the current date and presented audit by audit in the output file. An example of the generated output is presented below. (Figure 9.)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Timestamp**	**Averages**	**Case0**	**Case10**	**Case6**	**Case4**	**Case8**	**Case2**			**Internal Services**	-	
2	2010-01-01	1.6666667	10	0	0	0	0	0			2.5	0	
3	2010-02-01	1.6666667	10	0	0	0	0	0			2.5	0	
4	2010-03-01	1.6666667	10	0	0	0	0	0			2.5	0	
5	2010-04-01	1.6666667	10	0	0	0	0	0			2.5	0	
6	2010-05-01	3	10	8	0	0	0	0			4.5	0	
7	2010-06-01	2.5833333	7.5	8	0	0	0	0			3.875	0	
8	2010-07-01	2.5833333	7.5	8	0	0	0	0			3.875	0	
9	2010-08-01	2.5833333	7.5	8	0	0	0	0			3.875	0	
10	2010-09-01	2.5833333	7.5	8	0	0	0	0			3.875	0	
11	2010-10-01	2.5833333	7.5	8	0	0	0	0			3.875	0	
12	2010-11-01	2.5833333	7.5	8	0	0	0	0			3.875	0	
13	2010-12-01	2.5833333	7.5	8	0	0	0	0			3.875	0	
14	2011-01-01	2.5833333	7.5	8	0	0	0	0			3.875	0	
15	2011-02-01	2.5833333	7.5	8	0	0	0	0			3.875	0	
16	2011-03-01	5.6666667	7.5	8	9.5	9	0	0			8.5	0	
17	2011-04-01	5.6666667	7.5	8	9.5	9	0	0			8.5	0	
18	2011-05-01	6.8666667	7.5	7.7	9.5	9	7.5	0			8.425	3.75	
19	2011-06-01	5.85	7.2	7.4	4	9	7.5	0			6.9	3.75	
20	2011-07-01	6.1666667	6.9	7.1	4	9	10	0			6.75	5	
21	2011-08-01	6.0666667	6.6	6.8	4	9	10	0			6.6	5	
22	2011-09-01	5.9666667	6.3	6.5	4	9	10	0			6.45	5	
23	2011-10-01	6.8666667	6	6.2	4	9	10	6			6.3	8	

Figure 9. A sample of the output file

Additionally, graphs will be drawn to every sheet displaying the grades and averages in the time domain.

5.1.5 Flowchart

Flowchart is used to describe the flow of the script in a general level. The flowchart of the script is presented in the below figure. (Figure 10.) The execution is described respect to test activities.

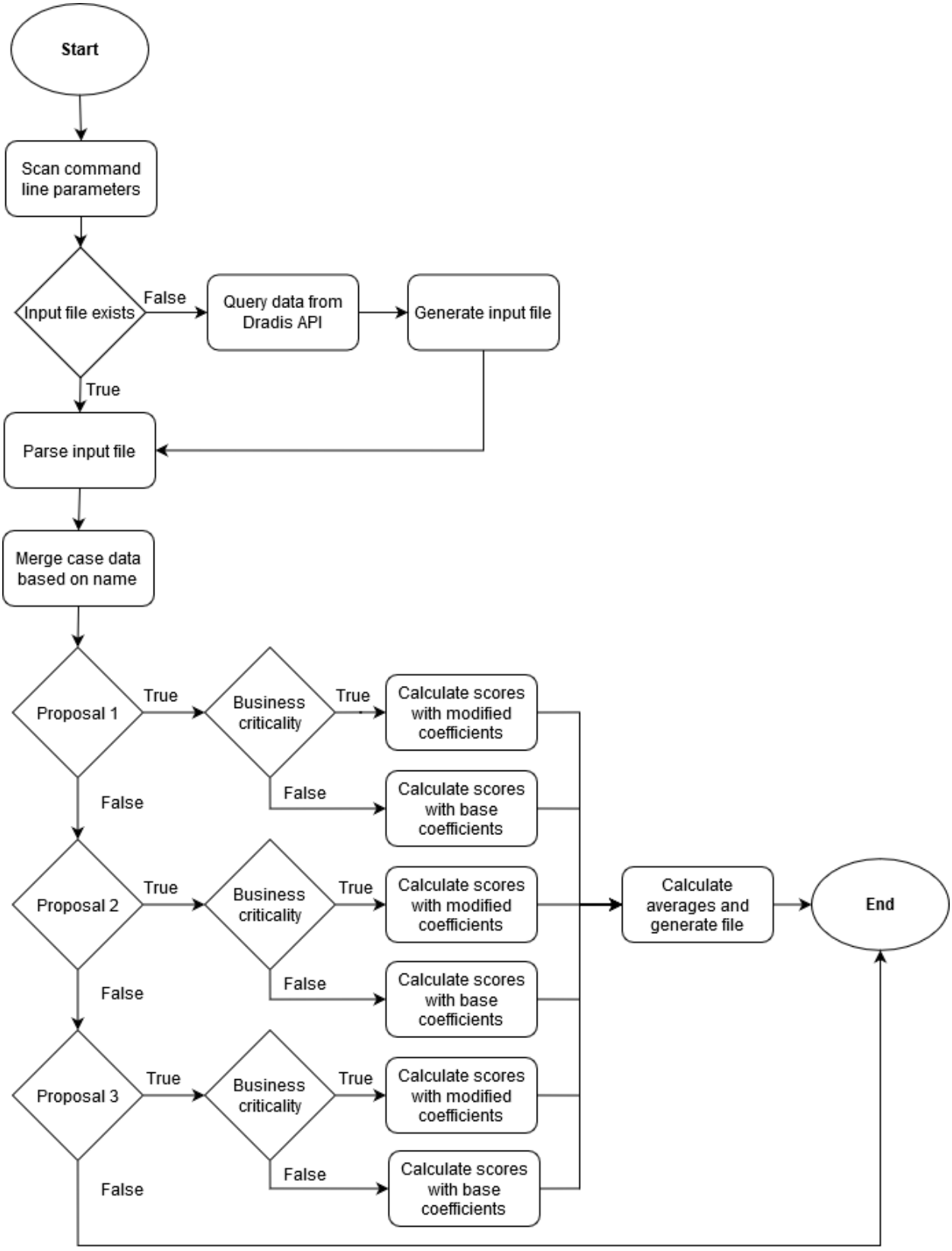

Figure 10. Flowchart

The execution of the script starts by scanning the command line parameters which include the selection of calculation proposal and whether the calculations should consider the business criticality or not. The selections are used later in the calculation process.

After the command line parameters have been scanned successfully, the script checks if the input file exists in the defined path. If the file does not exist, the script will establish a connection to the Dradis API, queries the cases and issues, masks the data and then generates the input file. On the other hand, if the input file does already exist the execution will continue to the parsing of the input file.

The input file can contain many entries of which are assigned to the same system or case with different timestamps. The described scenario can occur if the same system has multiple audits executed against them. Therefore, the cases are merged together during the script execution. The merging of the data is also done because it enables easier data handling during execution.

After the cases are merged, the script will calculate the security scores monthly according to the calculation proposal selected at the beginning of the execution. Additionally, it generates the scores and averages monthly for every group and category and for the whole environment and divides the scores based on the audit type. Finally, the script will write all the data to the excel workbook and generates the graphs.

5.2 Test case structures

Every calculation proposal will be tested against three different test cases which include tests with the base coefficients, with modified coefficients and with the weighted average. Therefore, the test case structure includes nine test cases in total. In this thesis, test case one corresponds the test with base coefficients, test case two corresponds the modified coefficient test and test case three corresponds the weighted average test. These correspondences are used interchangeably hereafter in this thesis.

The main objective of the tests is to examine and criticize the effects of different calculation proposals to the security score of the system. In addition, the objective is to investigate the effects of different coefficient values to security score by factoring in the business criticality of the system. Finally, the tests will provide data for the examination whether the business criticality should be considered in the whole environment or per entity or system level.

In order to compare results between different proposals, the same input file will be used for every test. Additionally, the input file entries follow the conditions which state that the order does matter and issue frequency can only vary between zero and three. This means that there are 24 different issue frequency combinations that can be presented in the file. Finally, the business criticality must be considered in the calculations and in order to compare the effect of business criticality to the systems security level, every case entry must contain the same issues frequencies and criticalities with different business criticality levels. Therefore, because there are three different levels, the input file will contain 72 case entries in total. In addition, the input file will contain three unknown systems to simulate the effect to the average in the environment contains corresponding systems. Because the calculations are the same for every case category, the input file will only contain cases categorized as technical audits. The input file used in testing is presented thoroughly in Appendix A.

6 RESULTS AND ANALYSIS

This chapter presents the test results. The proposals presented in the results are the same that were presented earlier (see chapter 4.4). Additionally, every test case will follow the guideline presented earlier in the thesis (see chapter 5.2) and the results for these cases will be presented using the raw data outputs and the graphs. The raw results and graphs will be used to analyze and compare together the test results in every test case. Additionally, the results will be criticized and compared to the already available frameworks. Finally, this chapter will present further recommendations, suggestions and development proposals to the developed security evaluation method and script.

6.1 Test case 1

In the first test case, the calculations are executed using the base coefficients which were presented in Table 2 in every proposal. In this test case the business criticality is not considered.

6.1.1 Proposal 1

The raw results for proposal 1 are presented below. (Table 6.)

Table 6. Raw results for proposal 1 in test case 1. The table contains security scores calculated for every timestamp. The security score can vary between zero and ten.

Timestamp	Averages	Case1	Case2	Case3
1.1.2017	3,333333333	5	5	0
1.2.2017	4,833333333	4,5	5	5
1.3.2017	4,5	4,5	4,5	4,5
1.4.2017	4,5	4,5	4,5	4,5
1.5.2017	4,166666667	3,5	4,5	4,5
1.6.2017	3,5	3,5	3,5	3,5
1.7.2017	3,5	3,5	3,5	3,5
1.8.2017	3,333333333	3	3,5	3,5
1.9.2017	3	3	3	3
1.10.2017	5,333333333	6,5	6,5	3
1.11.2017	6,333333333	6	6,5	6,5
1.12.2017	6	6	6	6
1.1.2018	7,333333333	8	8	6
1.2.2018	7,833333333	7,5	8	8
1.3.2018	7,5	7,5	7,5	7,5
1.4.2018	7,5	7,5	7,5	7,5
1.5.2018	6,833333333	6,5	6,5	7,5
1.6.2018	6,833333333	7,5	6,5	6,5
1.7.2018	7,5	7,5	7,5	7,5
1.8.2018	7,166666667	7	7	7,5
1.9.2018	6,833333333	6,5	7	7
1.10.2018	6,5	6,5	6,5	6,5
1.11.2018	5,833333333	5,5	5,5	6,5
1.12.2018	5,833333333	6,5	5,5	5,5
1.1.2019	6,5	6,5	6,5	6,5
1.2.2019	5,833333333	5,5	5,5	6,5
1.3.2019	5,666666667	6	5,5	5,5
1.4.2019	6	6	6	6
1.5.2019	4,666666667	4	4	6
1.6.2019	4,166666667	4,5	4	4
1.7.2019	4,5	4,5	4,5	4,5
1.8.2019	3,833333333	3,5	3,5	4,5
1.9.2019	3,5	3,5	3,5	3,5
1.10.2019	4,5	5	5	3,5
1.11.2019	4,833333333	4,5	5	5
1.12.2019	4,5	4,5	4,5	4,5
1.1.2020	2,25	4,5	4,5	4,5
1.2.2020	2,25	4,5	4,5	4,5

In addition, the following graph was generated from the data. (Figure 11.) The graph shows the effects of different vulnerability amounts to the security score when the calculations are done utilizing the proposal 1.

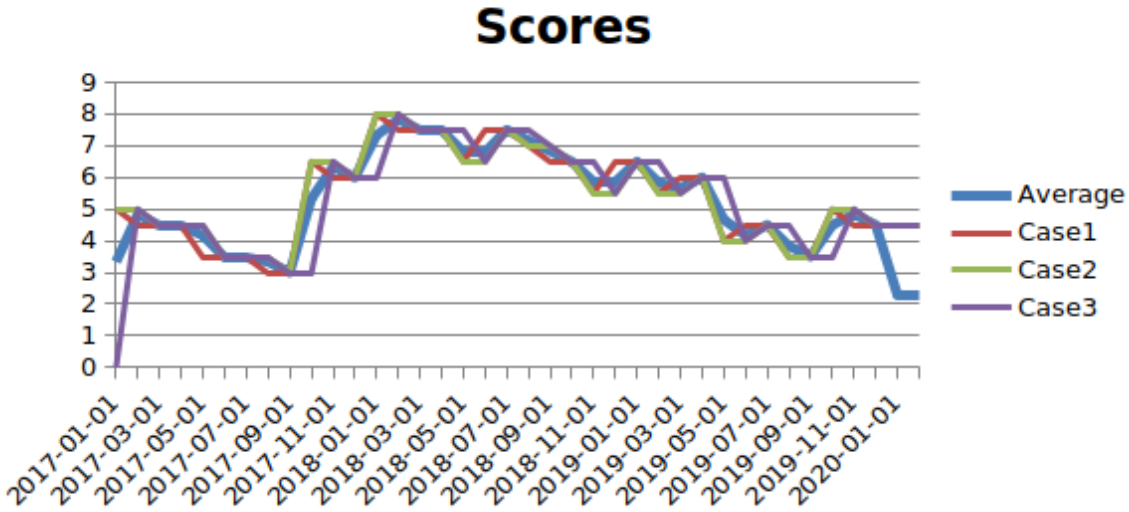

Figure 11. Graph generated from the results. The y-axis presents the security scores and the x-axis presents the timestamps.

By examining the above figure (Figure 11.), the raw data results (Table 6.) and the input file (Appendix A) it can be indicated that the more the case has critical vulnerabilities, the lower the security score will be and vice versa. In addition, the drop at the end of the graph originates from the fact the input file contains unknown systems which security score will be set to zero. This will affect the average in a major way.

6.1.2 Proposal 2

The raw results for proposal 2 are presented in the below table. (Table 7.)

Table 7. Raw results for proposal 2 in test case 1. The table contains security scores calculated for every timestamp. The security score can vary between zero and ten.

Timestamp	Averages	Case1	Case2	Case3
1.1.2017	4,8	7,2	7,2	0
1.2.2017	7,06	6,78	7,2	7,2
1.3.2017	6,78	6,78	6,78	6,78
1.4.2017	6,6	6,51	6,51	6,78
1.5.2017	6,216666667	5,63	6,51	6,51
1.6.2017	5,63	5,63	5,63	5,63
1.7.2017	5,403333333	5,29	5,29	5,63
1.8.2017	5,136666667	4,83	5,29	5,29
1.9.2017	4,83	4,83	4,83	4,83
1.10.2017	7,15	8,31	8,31	4,83
1.11.2017	8,17	7,89	8,31	8,31
1.12.2017	7,89	7,89	7,89	7,89
1.1.2018	8,656666667	9,04	9,04	7,89
1.2.2018	8,996666667	8,91	9,04	9,04
1.3.2018	8,91	8,91	8,91	8,91
1.4.2018	8,58	8,58	8,58	8,58
1.5.2018	8,213333333	8,03	8,03	8,58
1.6.2018	8,213333333	8,58	8,03	8,03
1.7.2018	8,58	8,58	8,58	8,58
1.8.2018	8,493333333	8,45	8,45	8,58
1.9.2018	8,18	7,64	8,45	8,45
1.10.2018	7,64	7,64	7,64	7,64
1.11.2018	7,273333333	7,09	7,09	7,64
1.12.2018	7,203333333	7,43	7,09	7,09
1.1.2019	7,43	7,43	7,43	7,43
1.2.2019	7,25	7,16	7,16	7,43
1.3.2019	7,09	6,95	7,16	7,16
1.4.2019	6,95	6,95	6,95	6,95
1.5.2019	6,183333333	5,8	5,8	6,95
1.6.2019	5,733333333	5,6	5,8	5,8
1.7.2019	5,6	5,6	5,6	5,6
1.8.2019	5,2	5	5	5,6
1.9.2019	5	5	5	5
1.10.2019	5,72	6,08	6,08	5
1.11.2019	6,033333333	5,94	6,08	6,08
1.12.2019	5,94	5,94	5,94	5,94
1.1.2020	2,97	5,94	5,94	5,94
1.2.2020	2,97	5,94	5,94	5,94

The graph generated from the input data is presented below. (Figure 12.) The graph shows the effects of different vulnerabilities to the security score when the calculations are done utilizing the proposal 2.

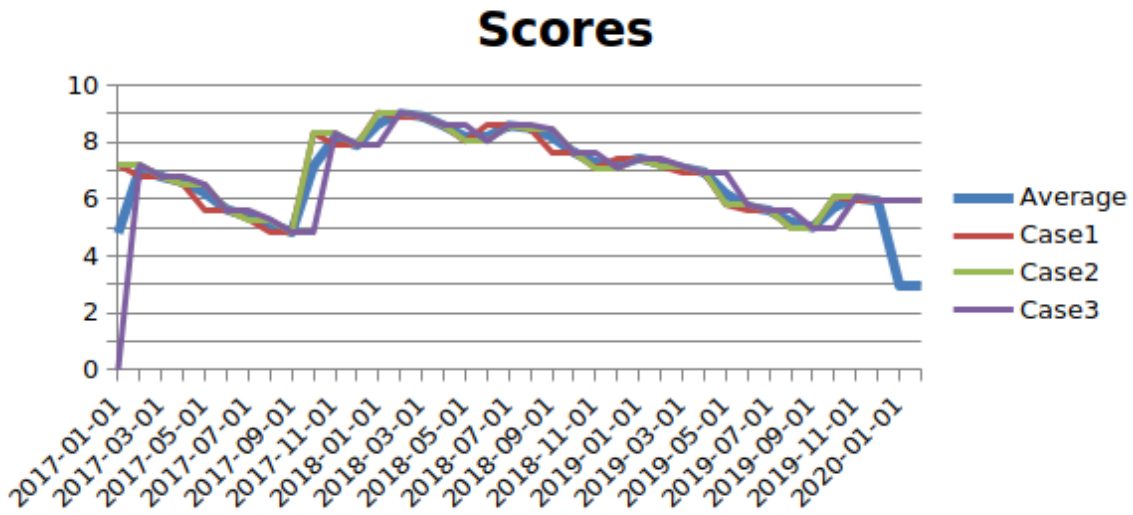

Figure 12. Graph generated from the results. The y-axis presents the security scores and the x-axis presents the timestamps.

By examining the above figure (Figure 12.), the raw data results (Table 7.) and the input file (Appendix A) it can be indicated that the effects of adding different vulnerabilities are not considerable because the differences between the security scores are not significant. This however does not concern the addition of critical vulnerabilities to the calculations.

6.1.3 Proposal 3

The results for proposal 3 are presented below. (Table 8.)

Table 8. Raw results for proposal 3 in test case 1. The table contains security scores calculated for every timestamp. The security score can vary between zero and ten.

Timestamp	Averages	Case1	Case2	Case3
1.1.2017	3,38	5,07	5,07	0
1.2.2017	4,336666667	2,87	5,07	5,07
1.3.2017	2,87	2,87	2,87	2,87
1.4.2017	3,11	3,23	3,23	2,87
1.5.2017	2,153333333	0	3,23	3,23
1.6.2017	0	0	0	0
1.7.2017	0	0	0	0
1.8.2017	0	0	0	0
1.9.2017	0	0	0	0
1.10.2017	4,12	6,18	6,18	0
1.11.2017	5,446666667	3,98	6,18	6,18
1.12.2017	3,98	3,98	3,98	3,98
1.1.2018	7,146666667	8,73	8,73	3,98
1.2.2018	8,496666667	8,03	8,73	8,73
1.3.2018	8,03	8,03	8,03	8,03
1.4.2018	7,33	7,33	7,33	7,33
1.5.2018	5,396666667	4,43	4,43	7,33
1.6.2018	5,71	8,27	4,43	4,43
1.7.2018	8,27	8,27	8,27	8,27
1.8.2018	7,803333333	7,57	7,57	8,27
1.9.2018	7,176666667	6,39	7,57	7,57
1.10.2018	6,39	6,39	6,39	6,39
1.11.2018	4,456666667	3,49	3,49	6,39
1.12.2018	4,003333333	5,03	3,49	3,49
1.1.2019	5,03	5,03	5,03	5,03
1.2.2019	4,263333333	3,88	3,88	5,03
1.3.2019	4,103333333	4,55	3,88	3,88
1.4.2019	4,55	4,55	4,55	4,55
1.5.2019	1,516666667	0	0	4,55
1.6.2019	0	0	0	0
1.7.2019	0	0	0	0
1.8.2019	0	0	0	0
1.9.2019	0	0	0	0
1.10.2019	0	0	0	0
1.11.2019	0	0	0	0
1.12.2019	0	0	0	0
1.1.2020	0	0	0	0
1.2.2020	0	0	0	0

The graph generated from the input data is presented below. (Figure 13.) The graph shows the effects of different vulnerability amounts to the security score when the calculations are done utilizing the proposal 3.

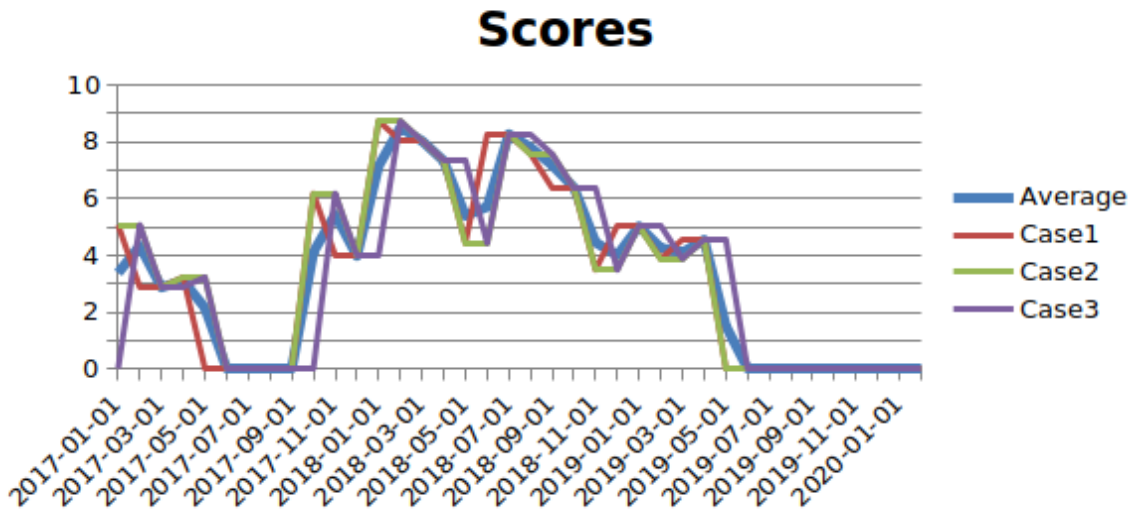

Figure 13. Graph generated from case data. The y-axis presents the security scores and the x-axis presents the timestamps.

By examining the above figure (Figure 13.), the raw data results (Table 8.) and the input file (Appendix A) it can be indicated that the effects of adding different vulnerabilities are significant and the security scores are varying considerably.

6.2 Test case 2

In the second test case, the calculations are executed using the modified coefficients which were presented in Table 3 in every proposal. In this test case, the business criticality is considered.

6.2.1 Proposal 1

The raw results for proposal 1 are presented below. (Table 9.)

Table 9. Raw results for proposal 1 in test case 2. The table contains security scores calculated for every timestamp. The security score can vary between zero and ten.

Timestamp	Averages	Case1	Case2	Case3
1.1.2017	2,733333333	3,8	4,4	0
1.2.2017	4,233333333	3,3	4,4	5
1.3.2017	3,9	3,3	3,9	4,5
1.4.2017	3,9	3,3	3,9	4,5
1.5.2017	3,566666667	2,3	3,9	4,5
1.6.2017	2,9	2,3	2,9	3,5
1.7.2017	2,9	2,3	2,9	3,5
1.8.2017	2,733333333	1,8	2,9	3,5
1.9.2017	2,4	1,8	2,4	3
1.10.2017	4,833333333	5,5	6	3
1.11.2017	5,833333333	5	6	6,5
1.12.2017	5,5	5	5,5	6
1.1.2018	7,033333333	7,4	7,7	6
1.2.2018	7,466666667	6,7	7,7	8
1.3.2018	7,1	6,7	7,1	7,5
1.4.2018	7,2	6,9	7,2	7,5
1.5.2018	6,433333333	5,7	6,1	7,5
1.6.2018	6,5	6,9	6,1	6,5
1.7.2018	7,2	6,9	7,2	7,5
1.8.2018	6,766666667	6,2	6,6	7,5
1.9.2018	6,5	5,9	6,6	7
1.10.2018	6,2	5,9	6,2	6,5
1.11.2018	5,433333333	4,7	5,1	6,5
1.12.2018	5,5	5,9	5,1	5,5
1.1.2019	6,2	5,9	6,2	6,5
1.2.2019	5,333333333	4,5	5	6,5
1.3.2019	5,3	5,4	5	5,5
1.4.2019	5,7	5,4	5,7	6
1.5.2019	4,166666667	3	3,5	6
1.6.2019	3,733333333	3,7	3,5	4
1.7.2019	4,1	3,7	4,1	4,5
1.8.2019	3,333333333	2,5	3	4,5
1.9.2019	3	2,5	3	3,5
1.10.2019	4,1	4,2	4,6	3,5
1.11.2019	4,366666667	3,5	4,6	5
1.12.2019	4	3,5	4	4,5
1.1.2020	2	3,5	4	4,5
1.2.2020	2	3,5	4	4,5

The graph generated from the input data is presented below. (Figure 14.) The graph shows the effects of different vulnerability amounts to the security score when the calculations are done considering the business criticality and utilizing proposal 1.

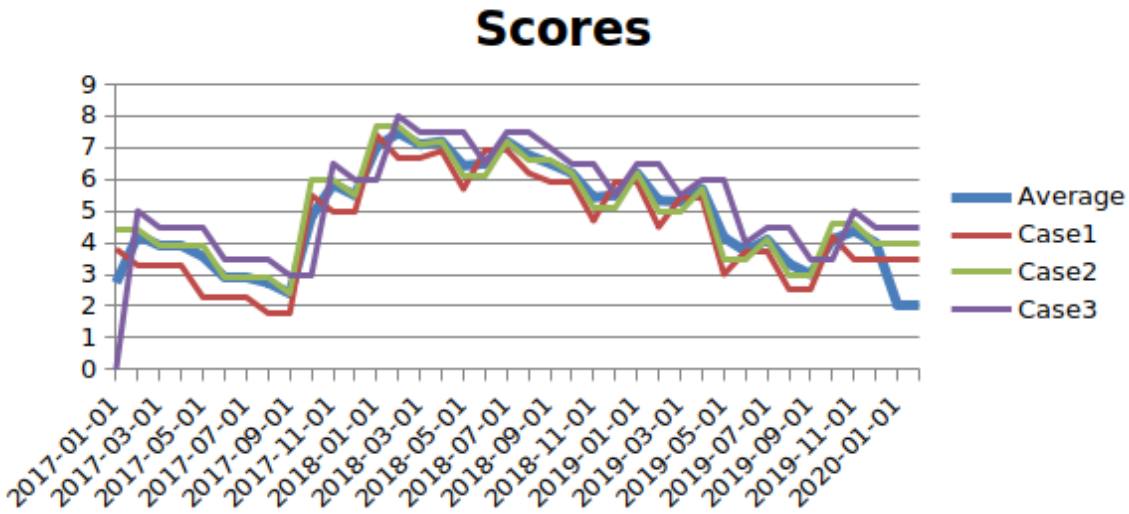

Figure 14. Graph generated from case data. The y-axis presents the security scores and the x-axis presents the timestamps.

By examining the above figure (Figure 14.), the raw data results (Table 9.) and the input file (Appendix A) it can be indicated that the effects of considering business criticality in the calculations will affect the security scores considerably.

6.2.2 Proposal 2

The results for proposal 2 are presented below. (Table 10.)

Table 10. Raw results for proposal 2 in test case 2. The table contains security scores calculated for every timestamp. The security score can vary between zero and ten.

Timestamp	Averages	Case1	Case2	Case3
1.1.2017	4,513	6,626	6,913	0
1.2.2017	6,753666667	6,148	6,913	7,2
1.3.2017	6,464	6,148	6,464	6,78
1.4.2017	6,287	5,884	6,197	6,78
1.5.2017	5,863	4,882	6,197	6,51
1.6.2017	5,256	4,882	5,256	5,63
1.7.2017	5,026333333	4,536	4,913	5,63
1.8.2017	4,738333333	4,012	4,913	5,29
1.9.2017	4,421	4,012	4,421	4,83
1.10.2017	6,937	7,884	8,097	4,83
1.11.2017	7,937666667	7,406	8,097	8,31
1.12.2017	7,648	7,406	7,648	7,89
1.1.2018	8,529666667	8,786	8,913	7,89
1.2.2018	8,852333333	8,604	8,913	9,04
1.3.2018	8,757	8,604	8,757	8,91
1.4.2018	8,421	8,262	8,421	8,58
1.5.2018	7,999333333	7,602	7,816	8,58
1.6.2018	8,051333333	8,308	7,816	8,03
1.7.2018	8,444	8,308	8,444	8,58
1.8.2018	8,331333333	8,126	8,288	8,58
1.9.2018	7,993333333	7,242	8,288	8,45
1.10.2018	7,441	7,242	7,441	7,64
1.11.2018	7,019333333	6,582	6,836	7,64
1.12.2018	6,989333333	7,042	6,836	7,09
1.1.2019	7,236	7,042	7,236	7,43
1.2.2019	7,002	6,664	6,912	7,43
1.3.2019	6,857333333	6,5	6,912	7,16
1.4.2019	6,725	6,5	6,725	6,95
1.5.2019	5,843333333	5,12	5,46	6,95
1.6.2019	5,41	4,97	5,46	5,8
1.7.2019	5,285	4,97	5,285	5,6
1.8.2019	4,825	4,25	4,625	5,6
1.9.2019	4,625	4,25	4,625	5
1.10.2019	5,436	5,512	5,796	5
1.11.2019	5,730666667	5,316	5,796	6,08
1.12.2019	5,628	5,316	5,628	5,94
1.1.2020	2,814	5,316	5,628	5,94

The graph generated from the input data is presented below. (Figure 15.) The graph shows the effects of different vulnerability amounts to the security score when the calculations are done considering the business criticality and utilizing proposal 2.

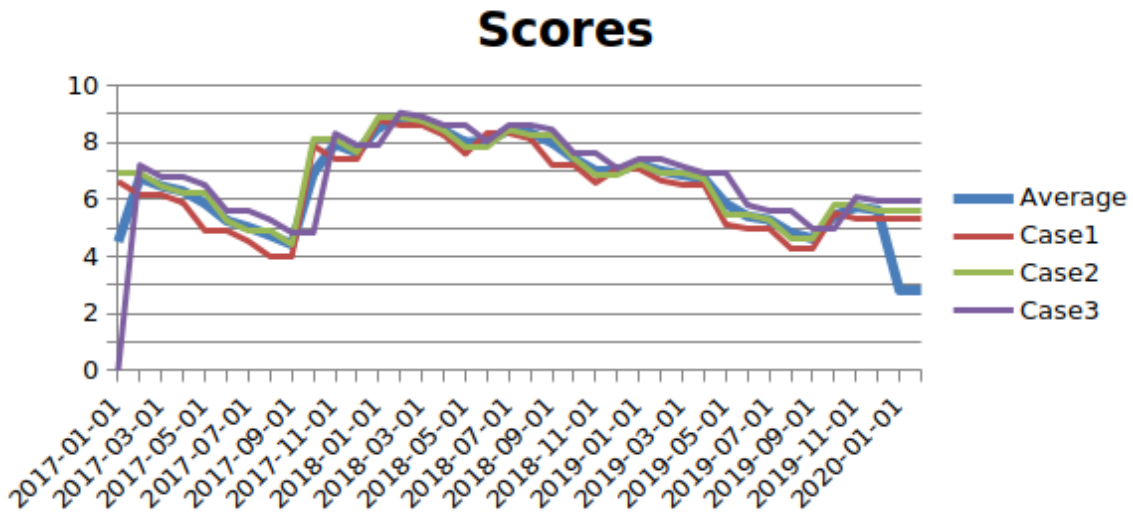

Figure 15. Graph generated from case data. The y-axis presents the security scores and the x-axis presents the timestamps.

By examining the above figure (Figure 15.), the raw data results (Table 10.) and the input file (Appendix A) it can be indicated that the effects of considering business criticality in the calculations will affect the security secure slightly.

6.2.3 Proposal 3

The results for proposal 3 are presented below. (Table 11.)

Table 11. Raw results for proposal 3 in test case 2. The table contains security scores calculated for every timestamp. The security score can vary between zero and ten.

Timestamp	Averages	Case1	Case2	Case3
1.1.2017	2,792	3,894	4,482	0
1.2.2017	3,648666667	1,394	4,482	5,07
1.3.2017	2,132	1,394	2,132	2,87
1.4.2017	2,461	1,932	2,581	2,87
1.5.2017	1,937	0	2,581	3,23
1.6.2017	0	0	0	0
1.7.2017	0	0	0	0
1.8.2017	0	0	0	0
1.9.2017	0	0	0	0
1.10.2017	3,606	5,152	5,666	0
1.11.2017	4,832666667	2,652	5,666	6,18
1.12.2017	3,316	2,652	3,316	3,98
1.1.2018	6,957666667	8,352	8,541	3,98
1.2.2018	8,214333333	7,372	8,541	8,73
1.3.2018	7,701	7,372	7,701	8,03
1.4.2018	7,046	6,762	7,046	7,33
1.5.2018	4,822666667	3,282	3,856	7,33
1.6.2018	5,386666667	7,874	3,856	4,43
1.7.2018	8,072	7,874	8,072	8,27
1.8.2018	7,465333333	6,894	7,232	8,27
1.9.2018	6,848	5,742	7,232	7,57
1.10.2018	6,066	5,742	6,066	6,39
1.11.2018	3,842666667	2,262	2,876	6,39
1.12.2018	3,562666667	4,322	2,876	3,49
1.1.2019	4,676	4,322	4,676	5,03
1.2.2019	3,679333333	2,712	3,296	5,03
1.3.2019	3,652	3,78	3,296	3,88
1.4.2019	4,165	3,78	4,165	4,55
1.5.2019	1,516666667	0	0	4,55
1.6.2019	0	0	0	0
1.7.2019	0	0	0	0
1.8.2019	0	0	0	0
1.9.2019	0	0	0	0
1.10.2019	0	0	0	0
1.11.2019	0	0	0	0
1.12.2019	0	0	0	0
1.1.2020	0	0	0	0
1.2.2020	0	0	0	0

The graph generated from the input data is presented below. (Figure 16.) The graph shows the effects of different vulnerability amounts to the security score when the calculations are done considering the business criticality and utilizing proposal 3.

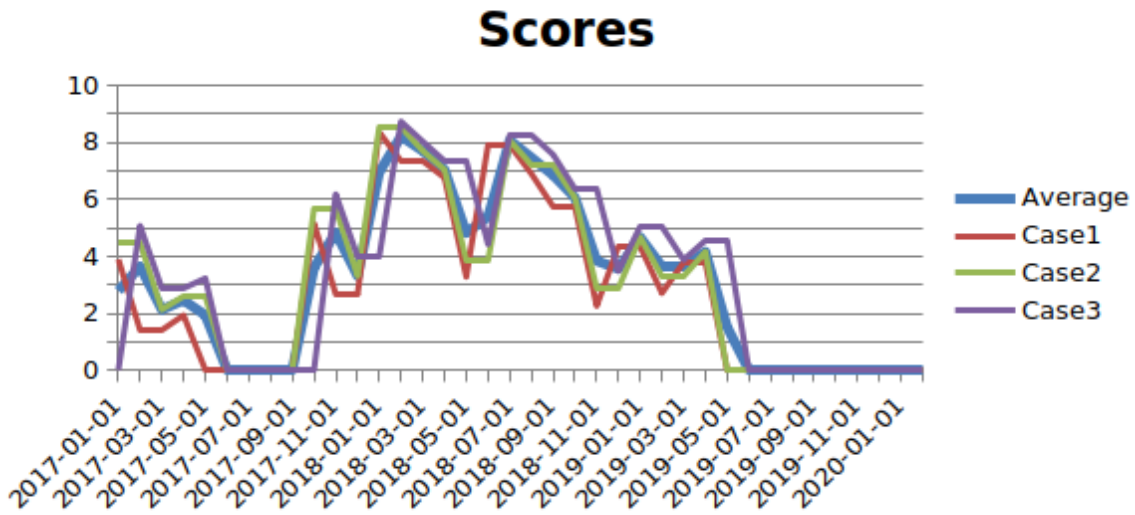

Figure 16. Graph generated from case data. The y-axis presents the security scores and the x-axis presents the timestamps.

By examining the above figure (Figure 16.), the raw data results (Table 11.) and the input file (Appendix A) it can be indicated that the effects of considering business criticality in the calculations will affect the security scores considerably.

6.3 Test case 3

The third test case is used for the weighted average calculations between every proposal. The results of this test case are utilized in the analysis section to determine whether the business criticality should be considered in the system or environment level.

6.3.1 Proposal 1

The results for proposal 1 are presented below. (Table 12.)

Table 12. Raw results for proposal 1 in test case 3. The table contains weighted and arithmetic averages calculated for every timestamp. The security score can vary between zero and ten.

Timestamp	Weighted average (all cases)	Average (all cases)
1.1.2017	3,888888889	3,333333333
1.2.2017	4,777777778	4,833333333
1.3.2017	4,5	4,5
1.4.2017	4,5	4,5
1.5.2017	4,055555556	4,166666667
1.6.2017	3,5	3,5
1.7.2017	3,5	3,5
1.8.2017	3,277777778	3,333333333
1.9.2017	3	3
1.10.2017	5,722222222	5,333333333
1.11.2017	6,277777778	6,333333333
1.12.2017	6	6
1.1.2018	7,555555556	7,333333333
1.2.2018	7,777777778	7,833333333
1.3.2018	7,5	7,5
1.4.2018	7,5	7,5
1.5.2018	6,722222222	6,833333333
1.6.2018	6,944444444	6,833333333
1.7.2018	7,5	7,5
1.8.2018	7,111111111	7,166666667
1.9.2018	6,777777778	6,833333333
1.10.2018	6,5	6,5
1.11.2018	5,722222222	5,833333333
1.12.2018	5,944444444	5,833333333
1.1.2019	6,5	6,5
1.2.2019	5,722222222	5,833333333
1.3.2019	5,722222222	5,666666667
1.4.2019	6	6
1.5.2019	4,444444444	4,666666667
1.6.2019	4,222222222	4,166666667
1.7.2019	4,5	4,5
1.8.2019	3,722222222	3,833333333
1.9.2019	3,5	3,5
1.10.2019	4,666666667	4,5
1.11.2019	4,777777778	4,833333333
1.12.2019	4,5	4,5
1.1.2020	2,7	2,25
1.2.2020	2,7	2,25

The graph generated from the input data is presented below. (Figure 17.) The graph shows the effects of proposal 1 to the average calculations.

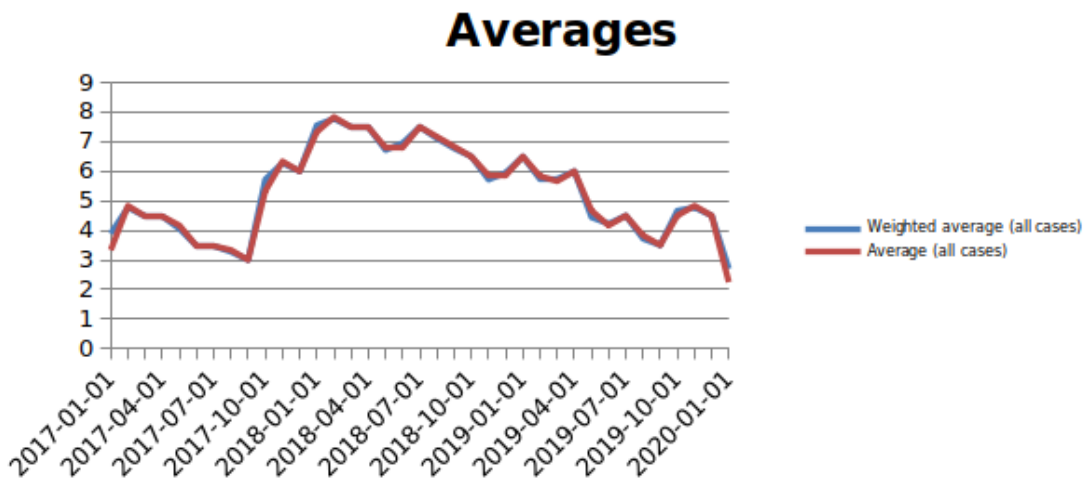

Figure 17. The graph generated from the output data. The y-axis presents the averages and the x-axis presents the timestamps.

By examining the above figure (Figure 17.), the raw data results (Table 12.) and the input file (Appendix A) it can be indicated that the weighted and arithmetic averages are almost the same in most cases. However, in some cases the weighted average is higher which indicates that there are more cases with high business criticality and security scores.

6.3.2 Proposal 2

The results for proposal 2 are presented below. (Table 13.)

Table 13. Raw results for proposal 2 in test case 3. The table contains weighted and arithmetic averages calculated for every timestamp. The security score can vary between zero and ten.

Timestamp	Weighted average (all cases)	Average (all cases)
1.1.2017	5,6	4,8
1.2.2017	7,013333333	7,06
1.3.2017	6,78	6,78
1.4.2017	6,57	6,6
1.5.2017	6,118888889	6,216666667
1.6.2017	5,63	5,63
1.7.2017	5,365555556	5,403333333
1.8.2017	5,085555556	5,136666667
1.9.2017	4,83	4,83
1.10.2017	7,536666667	7,15
1.11.2017	8,123333333	8,17
1.12.2017	7,89	7,89
1.1.2018	8,784444444	8,656666667
1.2.2018	8,982222222	8,996666667
1.3.2018	8,91	8,91
1.4.2018	8,58	8,58
1.5.2018	8,152222222	8,213333333
1.6.2018	8,274444444	8,213333333
1.7.2018	8,58	8,58
1.8.2018	8,478888889	8,493333333
1.9.2018	8,09	8,18
1.10.2018	7,64	7,64
1.11.2018	7,212222222	7,273333333
1.12.2018	7,241111111	7,203333333
1.1.2019	7,43	7,43
1.2.2019	7,22	7,25
1.3.2019	7,066666667	7,09
1.4.2019	6,95	6,95
1.5.2019	6,055555556	6,183333333
1.6.2019	5,711111111	5,733333333
1.7.2019	5,6	5,6
1.8.2019	5,133333333	5,2
1.9.2019	5	5
1.10.2019	5,84	5,72
1.11.2019	6,017777778	6,033333333
1.12.2019	5,94	5,94
1.1.2020	3,564	2,97
1.2.2020	3,564	2,97

The graph generated from the output data is presented below. (Figure 18.) The graph shows the effects of proposal 2 to the average calculations.

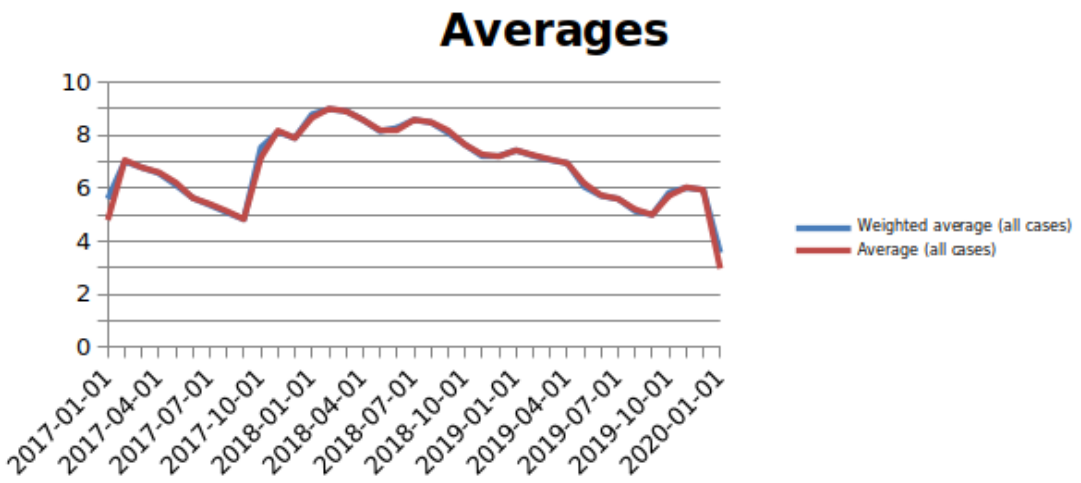

Figure 18. The graph generated from the output data. The y-axis presents the averages and the x-axis presents the timestamps.

By examining the above figure (Figure 18.), the raw data results (Table 13.) and the input file (Appendix A) it can be indicated that the weighted and arithmetic averages are almost the same in most cases. However, in some cases the weighted average will be higher than the arithmetic average and vice versa

6.3.3 Proposal 3

The results for proposal 3 are presented below. (Table 14.)

Table 14. Raw results for proposal 3 in test case 3. The table contains the weighted and arithmetic averages calculated for every timestamp. The security score can vary between zero and ten.

Timestamp	Weighted average (all cases)	Average (all cases)
1.1.2017	3,943333333	3,38
1.2.2017	4,092222222	4,336666667
1.3.2017	2,87	2,87
1.4.2017	3,15	3,11
1.5.2017	1,794444444	2,153333333
1.6.2017	0	0
1.7.2017	0	0
1.8.2017	0	0
1.9.2017	0	0
1.10.2017	4,806666667	4,12
1.11.2017	5,202222222	5,446666667
1.12.2017	3,98	3,98
1.1.2018	7,674444444	7,146666667
1.2.2018	8,418888889	8,496666667
1.3.2018	8,03	8,03
1.4.2018	7,33	7,33
1.5.2018	5,074444444	5,396666667
1.6.2018	6,136666667	5,71
1.7.2018	8,27	8,27
1.8.2018	7,725555556	7,803333333
1.9.2018	7,045555556	7,176666667
1.10.2018	6,39	6,39
1.11.2018	4,134444444	4,456666667
1.12.2018	4,174444444	4,003333333
1.1.2019	5,03	5,03
1.2.2019	4,135555556	4,263333333
1.3.2019	4,177777778	4,103333333
1.4.2019	4,55	4,55
1.5.2019	1,011111111	1,516666667
1.6.2019	0	0
1.7.2019	0	0
1.8.2019	0	0
1.9.2019	0	0
1.10.2019	0	0
1.11.2019	0	0
1.12.2019	0	0
1.1.2020	0	0
1.2.2020	0	0

The graph generated from the output data is presented below. (Figure 19.) The graph shows the effects of proposal 3 to the average calculations.

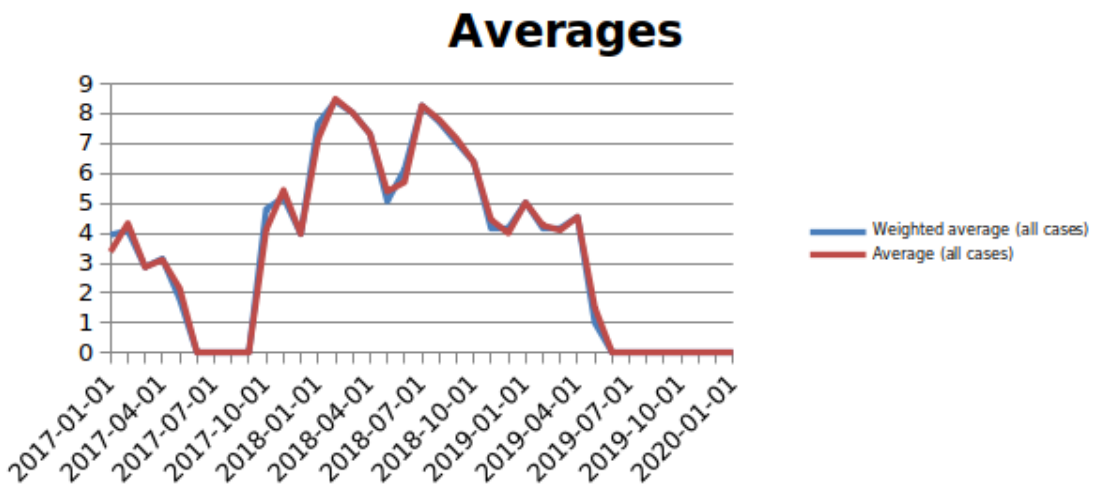

Figure 19. The graph generated from the output data. The y-axis presents the averages and the x-axis presents the timestamps.

By examining the above figure (Figure 19.), the raw data results (Table 14.) and the input file (Appendix A) it can be indicated that the weighted and arithmetic averages are almost the same in most cases.

6.4 Comparison and discussion between proposals

In this chapter, every test case will be compared together and criticized. The comparison process is divided into three different domains which are the comparison with base coefficient calculation results, comparison between base coefficient and modified coefficient calculations and the modified coefficient calculations comparison to weighted average. The results presented earlier for every test case and the input file will be used for the comparison activities.

6.4.1 With base coefficients

Calculations with base coefficients references the test case 1 in which every proposal was used to calculate the system security score using the coefficients described in Table 2. For easier comparison, the following table was constructed from the output data. (Table 15.) Entries for the table was selected from the output data so that it would present clearly the effects of adding distinct vulnerabilities to the calculations.

Table 15. Security scores gathered from the outputs. Score (1) references to proposal 1, Score (2) references to proposal 2 and Score (3) references to proposal 3

Criticals	Majors	Minors	Score (1)	Score (2)	Score (3)
-	6.5	3.4,2.8	8	9.4	8.73
-	6.5,6.0	3.4,2.8,2.6	6.5	8.31	6.39
-	6.5,6.0,5.5	3.4,2.8	6	7.89	3.98
7.4	6.5,6.0	3.4,2.8,2.6	5	7.2	5.07
7.4,8.6	6.5,6.0,5.5	3,4	3.5	5.63	0
7.4,8.6,9.0	6.5,6.0	3,4	3	4.83	0

By examining the above table (Table 15.), it can be clearly indicated that in most cases by adding more severe vulnerabilities to the calculations, the security score of the system will drop. However, in the third score column which represents the third proposal, the security score drops dramatically after adding one major vulnerability to the column. This behaviour is expected to some extent but the security score then raises again after one critical vulnerability is added and one major vulnerability is removed from the calculations. Expected behaviour in that case should be that the security score would be lower or at least the close to the same value as in the previous calculations.

Another assumption in this case would be that the presence of critical vulnerabilities should lower the score significantly. This assumption is fulfilled partly by the first and second proposal. However, the second proposals score appears too permissive compared to the amount of critical vulnerabilities in the system. The first proposal appears to fulfil this assumption the best compared to other proposals.

6.4.2 Base versus modified coefficients

Modified coefficients were used in the calculations to factor in business criticality of the target system. The higher the business criticality, the more it will affect the final score as a lowering factor. The following table contains data from all the proposals and scores which have also business criticalities factored in the calculations (Table 16.) Entries in the table have been selected respecting the same principle as in base coefficient effect comparison.

Table 16. Security scores gathered from the outputs. Score (1) references to proposal 1, Score (2) references to proposal 2 and Score (3) references to proposal 3

Business criticality	Criticals	Majors	Minors	Score (1)	Score (2)	Score (3)
normal	-	6.5	3.4,2.8	8	9.4	8.73
moderate	-	6.5	3.4,2.8	7.7	8.913	8.541
high	-	6.5	3.4,2.8	7.4	8.786	8.352
normal	-	6.5,6.0	3.4,2.8,2.6	6.5	8.31	6.39
moderate	-	6.5,6.0	3.4,2.8,2.6	6.2	8.097	5.666
high	-	6.5,6.0	3.4,2.8,2.6	5.9	7.884	5.152
normal	-	6.5,6.0,5.5	3.4,2.8	6	7.89	3.98
moderate	-	6.5,6.0,5.5	3.4,2.8	5.5	7.648	3.316
high	-	6.5,6.0,5.5	3.4,2.8	5	7.406	2.652
normal	7.4	6.5,6.0	3.4,2.8,2.6	5	7.2	5.07
moderate	7.4	6.5,6.0	3.4,2.8,2.6	4.4	6.913	4.482
high	7.4	6.5,6.0	3.4,2.8,2.6	3.8	6.626	3.894
normal	7.4,8.6	6.5,6.0,5.5	3,4	3.5	5.63	0
moderate	7.4,8.6	6.5,6.0,5.5	3,4	2.9	5.256	0
high	7.4,8.6	6.5,6.0,5.5	3,4	2.3	4.882	0
normal	7.4,8.6,9.0	6.5,6.0	3,4	3	4.83	0
moderate	7.4,8.6,9.0	6.5,6.0	3,4	2.4	4.412	0
high	7.4,8.6,9.0	6.5,6.0	3,4	1.8	4.012	0

Like in the base coefficient comparison, it can be clearly indicated from the above table (Table 16.) that the security scores are lowering whenever more severe vulnerabilities are added in most of the proposals. In this case, the business criticality will also affect the

security score of the system as a modifying factor. By adding the business criticality to the calculations, the score will offer a more precise picture of the security state of the system and environment. This is because in a case where a highly business critical banking system which has the same vulnerabilities as an application which does not contain business critical data, should have a lower security score because of the nature and criticality of the data it stores and handles. Additionally, if the application has critical vulnerabilities, it usually means that the confidentiality, integrity and availability of the data are at risk and in an application which handles and stores sensitive data the consequences of potential exploit and compromise can be more severe.

6.4.3 Modified coefficients versus weighted average

Modified coefficients and weighted average were used in the calculations to offer solutions and alternatives on how to consider the business criticality of the system. The modified coefficient solution considers the business criticality in system level and the weighted average considers the business criticality in the environment level. Tables and figures presented in the test case 2 and 3 were used in the comparison process.

Both, the modified coefficient and weighted average consider the business criticality of the system which is beneficial in the security state description. However, the weighted average describes better the whole environments security level because it weights distinct systems with different weights based on the business criticality of the system and considers them in the average calculations. On the other hand, if the systems security scores are calculated without using the modified coefficients it can offer a corrupted view of the real security state of a single system. This can be misleading if the scores are examined in the system level. Thus, the calculations should be done on the system level with modified coefficients. However, the weighted average could still be added to the calculations to offer a more specify description of the whole environments state.

6.4.4 Criticism and comparison to available frameworks

Although one of the requirements was to develop the security evaluation method as simple as possible, the end result can be too simple and straightforward for some cases. The already available frameworks contain many levels and categories which can be used to evaluate the systems and environments security state. Although in some cases the process can become more time consuming by adding more layers, all of the layers are implemented for a reason in these frameworks. Thus, the already available frameworks are recommended if the system or environment demands a more extended approach in terms of security.

The unknown system considerations in the calculations may be too strict and harsh for some situations. This can lead to misleading information regarding the true security state of the system and the whole environment. This can happen for example when some third-party has audited the system and the results of the audit are not known.

The evaluation process does not categorize distinct issues. For example, in the NIST cyber security framework the issues are categorized into different domains such as asset and identity management. The approach could be used also in the developed security evaluation process to provide a more extended approach to the security of the system and the whole environment. This information could also be used to detect weaknesses in the development process of the application or system.

The CVSS scores contain four different levels which are minor, major, high and critical. This approach could also be used in the new security state evaluation process to better address the severity of the vulnerability which is present in the system to the organization and the effects of the vulnerability to the security score. Occasionally, in a case where CVSS score is distinctly over the critical threshold used in the new security evaluation process, the severity and consequences of exploiting the vulnerability may be clearly higher. In these cases, these vulnerabilities could be weighted more by implementing a fourth level in to the new security evaluation process.

Finally, new technologies and vulnerabilities are emerging every day and therefore any digital system can never be interpreted as fully secure. This also applies to the systems which security is evaluated with audits and with the new security evaluation process.

7 CONCLUSIONS

The objective of the study was to develop a security evaluation process which would be easy for small and medium-sized enterprises (SME) to implement and utilize in their continuous information or cyber security processes. In addition, the objective was to develop a script which automates the calculations and can be used in testing. These objectives were fulfilled during the study. The designed information security evaluation method was compared and criticized against two existing security frameworks which are the NIST (National Institute of Standards and Technology) cyber security framework and CIS (Center for Information Security) Critical Security Controls. Finally, the new design was tested against three different test cases in order to verify the correct functioning of the new security evaluation method and to compare the proposals together. The results are mainly applicable for the company who ordered the thesis because the new security evaluation method was designed based on the requirements stated by them. However, if an another principal wants to evaluate the security state of a system and has the same requirements for the evaluation process, the developed evaluation method can possibly be applied to that companies' processes also.

The study included three different proposals which were constructed according to audit results. The features that were taken into account in the proposals were the frequency of issues per vulnerability rating and the CVSS scores. In addition, three test cases were implemented for testing. These test cases included tests with base coefficients, with modified coefficients and the weighted average calculations. The implemented test cases and the results were compared in three distinct scenarios which were the comparison between proposals with base coefficients, modified coefficients and with the weighted average. The results indicated that the best option for the evaluation process was the first proposal with modified coefficients and with weighted average.

When considering all the discussed proposals, results and facts, the best option right now is to implement the proposal 1 with business criticality and weighted average calculations. The mentioned proposal appeared to consider best the effects of adding different vulnerabilities to the system. In addition, by adding the business criticality and weighted average

to the proposal, the calculations would describe better the state of security in the system and in the environment. Although this proposal was valued as the best solution right now, in the future it is recommended to examine the effects of different coefficients and modify them in order to achieve a more comprehensive description of the security state.

Additionally, in order to perfect the calculation process, the calculations should be continuously developed. One feasible development proposal could be that the new security evaluation process could also be developed and approached so that the input would also contain information about the target system and not only audit results. By organizing the input so, the security of the system could be evaluated based on the operations and actions that have been executed against the system. These operations which would be considered in the evaluation process could be for example technical audit, technical review, incident response plan and so on. For instance, if a technical audit has only been executed against the system, the security score would be affected negatively because other important security operations and actions have not been executed. This approach would offer a more extended outlook of the security state.

Finally, the new security evaluation process could be integrated as a part of a web service which could parse the audit data and calculate the target systems security level automatically in the backend and display the security states in the client side. By implementing the security evaluation process as a part of web service, it would offer scalability, and the target systems security state evaluation could be implemented so that it would automatically update based on different features such as time.

REFERENCES

Bartock, M. & Cichonski, J. & Souppaya, M. & Smith, M. & Witte, G. & Scarfone, K. (2016). Guide for Cybersecurity Event Recovery. [online] National Institute of Standards and Technology. NIST Special Publication 800-184. [Referenced 28.12.2019] Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-184.pdf

Bonfietti, A. & Lombardi, M. (2012). The Weighted Average Constraint. [online]. DEIS, University of Bologna. [Referenced 16.3.2020] Available: https://www.researchgate.net/publication/262240369_The_Weighted_Average_Constraint

Bowen, P. Hash, J. Wilson, M. (2006). Information Security Handbook: A Guide for Managers. [online]. Computer Security Division. National Institute of Standards and Technology. [Referenced 23.12.2019] Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-100.pdf

CIS (a). (2017). MS-ISAC Security Primer: Cross-Site Scripting (XSS). [online] [Referenced 28.12.2019] Available: https://www.cisecurity.org/wp-content/uploads/2015/07/Security-Primer-XSS-1.pdf

CIS (b). (2019). About us. [online] Center for Internet Security. [Referenced 31.12.2019] Available: https://www.cisecurity.org/about-us/

CIS (c). (2019). CIS Controls. [online] Center for Internet Security. [Referenced 5.1.2020] Available: https://www.cisecurity.org/controls/

Fasulo, P. (2019). Cybersecurity vs Information Security: What is the difference? [online] [Referenced 23.12.2019] Available: https://securityscorecard.com/blog/information-security-versus-cybersecurity

Forum of Incident Response Security Teams (FIRST). (2019). Common Vulnerability Scoring System version 3.1, Specification Document, Revision 1. [online] [Referenced 13.3.2020] Available: https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf

Ghazouni, M. & Medormi, H. & Boulafdour, B. & Sayouti, A. (2013). A Model for an information security management system (ISMS Tool) based multi agent system. [online] International Conference on Intelligent Information and Network Technology (IC2INT'13). [Referenced 13.3.2020] Available: https://www.researchgate.net/publication/292140170_A_model_for_an_Information_security_management_system_ISMS_Tool_based_multi_agent_system

Golyash, I. & Sachenko, S. & Rippa, S. (2011). Improving the Information Security Audit of Enterprise Using XML Technologies. [online] Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, Prague, 2011, pp. 795-798. [Referenced 26.3.2020]. Available: https://ieeexplore.ieee.org/document/6072879

Gupta, N. & Jain, A. & Saini, P. & Gupta, V. (2016). DDoS attack algorithm using ICMP flood. [online] 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp. 4082-4084. [Referenced 26.3.2020] Available: https://ieeexplore.ieee.org/document/7725026

Kaspersky labs. (2019). What is cyber security? [online] [Referenced 23.12.2019] Available: https://usa.kaspersky.com/resource-center/definitions/what-is-cyber-security

Katole, R. & Sherekar, S. & Thakare, V. (2018). Detection of SQL injection attacks by removing the parameter values of SQL query. [online] 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, 2018, pp. 736-741. [Referenced 28.3.2020] Available: https://ieeexplore.ieee.org/document/8398896

Ko, M. & Osei-Bryson, K-M. & Dorantes, A. (2006). Investigating the Impact of Publicly Announced Information Security Breaches on Three Performance Indicators of the Breached Firms. [online] Department of Information Systems and Technology Management. College of Business. The University of Texas at San Antonio. [Referenced 24.12.2019] Available: https://www.researchgate.net/publication/220121692_Investigating_the_Impact_of_Publicly_Announced_Information_Security_Breaches_on_Three_Performance_Indicators_of_the_Breached_Firms

Lutz, M. (2009). Learning Python, Fourth Edition. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. [Referenced 17.3.2020]

Mudarri, T. & Al-Rabeei, S A. (2015). Security Fundamentals: Access Control Models. [online] International Journal of Interdisciplinarity in Theory and Practice. ITPB – NR: 7. ISSN 2344-2409. [Referenced 28.12.2019] Available: https://www.researchgate.net/publication/282219117_SECURITY_FUNDAMENTALS_ACCESS_CONTROL_MODELS

National Cyber Security Center. (2018). Information Security in 2018. [online] Traficom. Finnish Transport and Communications Agency. [Referenced 28.12.2019] Available: https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/publication/Vuosikatsaus_2018_EN.pdf

Nieles, M. & Dempsey, K. & Yan Pillitteri, V. (2017). An Introduction to Information Security. [online]. National Institute of Standards and Technology. [Referenced 23.12.2019] Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-12r1.pdf

NIST (a). (2018). Framework for Improving Critical Infrastructure Cybersecurity. [online] [Referenced 8.12.2019] Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf

NIST (b). (2018). An Introduction to the Components of the Framework. [online] National Institute of Standards and Technology. [Referenced 29.12.2019] Available: https://www.nist.gov/cyberframework/online-learning/components-framework

Oscarson, P. Information Security Fundamentals. [online] Department of Business Administration, Economics, Statistics and Informatics, Örebro University, Sweden. [Referenced 24.12.2019] Available: https://pdfs.semanticscholar.org/ae28/9631332e66616044df36797ca71385f204ab.pdf

OWASP (a). (2017). OWASP top 10: A1 Injection. [online] [Referenced 27.12.2019] Available: https://www.owasp.org/index.php/Top_10-2017_A1-Injection

OWASP (b). (2017). OWASP Top 10: A7 Cross-Site Scripting. [online] [Referenced 27.12.2019] Available: https://www.owasp.org/index.php/Top_10-2017_A7-Cross-Site_Scripting_(XSS)

OWASP (c). (2017). OWASP Top 10: Using components with known vulnerabilities. [online] Open Web Application Security Project. [Referenced 28.12.2019] Available: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

Paulsen, C. & Boyens, J. & Bartol, N. & Winkler, K. (2018). Criticality Analysis Process Model: Prioritizing Systems and Components. [online] National Institute of Standards and Technology (NIST). [Referenced 13.3.2020] Available: https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8179.pdf

Paulsen, C. & Byers, R. (2019). Glossary of Key Information Security Terms. [online]. Computer Security Division. National Institute of Standards and Technology. [Referenced 23.12.2019] Available: https://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf

Pender-Bey, G. (2012). The Parkerian Hexad: The CIA Expanded. [online] Lewis University. [Referenced 24.12.2019] Available: https://cs.lewisu.edu/mathcs/msisprojects/papers/georgiependerbey.pdf

Pranathi, K. & Kranthi, S. & Srisaila, A. & Madhavilatha, P. (2018). Attacks on Web Application Caused by Cross Site Scripting. [online] Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 2018, pp. 1754-1759. [Referenced 28.3.2020] Available: https://ieeexplore.ieee.org/document/8474765

Reetz, S. (2017). Technical White Paper: SQL Injection. [online] Multi-State Information Sharing and Analysis Center (MS-ISAC). Center for Internet Security. [Referenced 27.12.2019] Available: https://www.cisecurity.org/wp-content/uploads/2017/05/SQL-Injection-White-Paper2.pdf

Rouse, M. (2015). Framework. [online] [Referenced 29.12.2019] Available: https://whatis.techtarget.com/definition/framework

Security Roots Ltd. (2020). Dradis Framework. [online] [Referenced 17.3.2020] Available: https://dradisframework.com/

Soliman, M. & Azer, M. (2018). Web Application API Blind Denial of Service Attacks. [online]. 14th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 2018, pp. 249-253. [Referenced 26.3.2020] Available: https://ieeexplore.ieee.org/document/8636115

Stine, K. & Dang, Q. Encryption Basics. [online] National Institute of Standards and Technology Information, Technology Laboratory, Computer Security Division. [Referenced 28.12.2019] Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=908084

Thurimella, R. & Mitchell, W. (2009). Cloak and Dagger: Man-In-The-Middle and Other Insidious Attacks. [online] University of Denver. International Journal of Information Security and Privacy. [Referenced 28.12.2019] Available: https://www.cs.du.edu/~ramki/downloads/papers/cloakPreprint.pdf

Von Solms, B. (2001). Corporate Governance and Information Security. [online] [Referenced 8.12.2019] Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.1168&rep=rep1&type=pdf

Von Solms, R. & van Niekerk. J. (2013). From information security to cyber security. [online]. School of ICT, Nelson Mandela Metropolitan University, Port Elizabeth 6031, South Africa. [Referenced 23.12.2019] Available: https://profsandhu.com/cs5323_s18/Solms-Niekerk-2013.pdf

APPENDIX A

Case	Type	Timestamp	Criti-cals/High	Majors/Me-dium	Mi-nors/Low	Audited	Critical-ity
Case1	audit	1.1.2017	7,4	6.5,6.0	3.4,2.8,2.6	1	high
Case2	audit	1.1.2017	7,4	6.5,6.0	3.4,2.8,2.6	1	moder-ate
Case3	audit	1.2.2017	7,4	6.5,6.0	3.4,2.8,2.6	1	normal
Case1	audit	1.2.2017	7,4	6.5,5.5,6.0	3.4,2.8	1	high
Case2	audit	1.3.2017	7,4	6.5,5.5,6.0	3.4,2.8	1	moder-ate
Case3	audit	1.3.2017	7,4	6.5,5.5,6.0	3.4,2.8	1	normal
Case1	audit	1.4.2017	7.4,8.6	6,5	3.4,2.8,2.6	1	high
Case2	audit	1.4.2017	7.4,8.6	6,5	3.4,2.8,2.6	1	moder-ate
Case3	audit	1.5.2017	7.4,8.6	6,5	3.4,2.8,2.6	1	normal
Case1	audit	1.5.2017	7.4,8.6	6.5,6.0,5.5	3,4	1	high
Case2	audit	1.6.2017	7.4,8.6	6.5,6.0,5.5	3,4	1	moder-ate
Case3	audit	1.6.2017	7.4,8.6	6.5,6.0,5.5	3,4	1	normal
Case1	audit	1.7.2017	7.4,8.6,9.0	6,5	3.4,2.8	1	high
Case2	audit	1.7.2017	7.4,8.6,9.0	6,5	3.4,2.8	1	moder-ate
Case3	audit	1.8.2017	7.4,8.6,9.0	6,5	3.4,2.8	1	normal
Case1	audit	1.8.2017	7.4,8.6,9.0	6.5,6.0	3,4	1	high
Case2	audit	1.9.2017	7.4,8.6,9.0	6.5,6.0	3,4	1	moder-ate
Case3	audit	1.9.2017	7.4,8.6,9.0	6.5,6.0	3,4	1	normal
Case1	audit	1.10.2017		6.5,6.0	3.4,2.8,2.6	1	high
Case2	audit	1.10.2017		6.5,6.0	3.4,2.8,2.6	1	moder-ate
Case3	audit	1.11.2017		6.5,6.0	3.4,2.8,2.6	1	normal
Case1	audit	1.11.2017		6.5,6.0,5.5	3.4,2.8	1	high
Case2	audit	1.12.2017		6.5,6.0,5.5	3.4,2.8	1	moder-ate
Case3	audit	1.12.2017		6.5,6.0,5.5	3.4,2.8	1	normal
Case1	audit	1.1.2018		6,5	3.4,2.8	1	high
Case2	audit	1.1.2018		6,5	3.4,2.8	1	moder-ate
Case3	audit	1.2.2018		6,5	3.4,2.8	1	normal
Case1	audit	1.2.2018		6,5	3.4,2.8,2.6	1	high
Case2	audit	1.3.2018		6,5	3.4,2.8,2.6	1	moder-ate
Case3	audit	1.3.2018		6,5	3.4,2.8,2.6	1	normal
Case1	audit	1.4.2018		6.5,6.0	3,4	1	high

Case2	audit	1.4.2018		6.5,6.0	3,4	1	moder-ate
Case3	audit	1.4.2018		6.5,6.0	3,4	1	normal
Case1	audit	1.5.2018		6.5,6.0,5.5	3,4	1	high
Case2	audit	1.5.2018		6.5,6.0,5.5	3,4	1	moder-ate
Case3	audit	1.6.2018		6.5,6.0,5.5	3,4	1	normal
Case1	audit	1.6.2018	7,4		3.4,2.8	1	high
Case2	audit	1.7.2018	7,4		3.4,2.8	1	moder-ate
Case3	audit	1.7.2018	7,4		3.4,2.8	1	normal
Case1	audit	1.8.2018	7,4		3.4,2.8,2.6	1	high
Case2	audit	1.8.2018	7,4		3.4,2.8,2.6	1	moder-ate
Case3	audit	1.9.2018	7,4		3.4,2.8,2.6	1	normal
Case1	audit	1.9.2018	7,4	6.5,6.0		1	high
Case2	audit	1.10.2018	7,4	6.5,6.0		1	moder-ate
Case3	audit	1.10.2018	7,4	6.5,6.0		1	normal
Case1	audit	1.11.2018	7,4	6.5,6.0,5.5		1	high
Case2	audit	1.11.2018	7,4	6.5,6.0,5.5		1	moder-ate
Case3	audit	1.12.2018	7,4	6.5,6.0,5.5		1	normal
Case1	audit	1.12.2018	7.4,8.6		3,4	1	high
Case2	audit	1.1.2019	7.4,8.6		3,4	1	moder-ate
Case3	audit	1.1.2019	7.4,8.6		3,4	1	normal
Case1	audit	1.2.2019	7.4,8.6		3.4,2.8,2.6	1	high
Case2	audit	1.2.2019	7.4,8.6		3.4,2.8,2.6	1	moder-ate
Case3	audit	1.3.2019	7.4,8.6		3.4,2.8,2.6	1	normal
Case1	audit	1.3.2019	7.4,8.6	6,5		1	high
Case2	audit	1.4.2019	7.4,8.6	6,5		1	moder-ate
Case3	audit	1.4.2019	7.4,8.6	6,5		1	normal
Case1	audit	1.5.2019	7.4,8.6	6.5,6.0,5.5		1	high
Case2	audit	1.5.2019	7.4,8.6	6.5,6.0,5.5		1	moder-ate
Case3	audit	1.6.2019	7.4,8.6	6.5,6.0,5.5		1	normal
Case1	audit	1.6.2019	7.4,8.6,9.0	6,5		1	high
Case2	audit	1.7.2019	7.4,8.6,9.0	6,5		1	moder-ate
Case3	audit	1.7.2019	7.4,8.6,9.0	6,5		1	normal
Case1	audit	1.8.2019	7.4,8.6,9.0	6.5,6.0		1	high
Case2	audit	1.8.2019	7.4,8.6,9.0	6.5,6.0		1	moder-ate

Case3	audit	1.9.2019	7.4,8.6,9.0	6.5,6.0		1	normal
Case1	audit	1.10.2019	7.4,8.6,9.0		3,4	1	high
Case2	audit	1.10.2019	7.4,8.6,9.0		3,4	1	moder-ate
Case3	audit	1.11.2019	7.4,8.6,9.0		3,4	1	normal
Case1	audit	1.11.2019	7.4,8.6,9.0		3.4,2.8	1	high
Case2	audit	1.12.2019	7.4,8.6,9.0		3.4,2.8	1	moder-ate
Case3	audit	1.12.2019	7.4,8.6,9.0		3.4,2.8	1	normal
Case10	audit	1.1.2020				0	
Case11	audit	1.1.2020				0	
Case12	audit	1.1.2020				0	