LUNDS UNIVERSITET
Ekonomihögskolan

Institutionen för informatik

Securing Sweden's Digital Assets

A retrospective analysis of Database Security in Sweden, with insights from top security experts.

Bachelor thesis 15hp, course SYSK16 in Information Systems

Authors: Per Hafström Fremlin
Daniel Tomic Lindvall

Supervisor: Blerim Emruli

Examiners: Benjamin Weaver
Umberto Fiaccadori

Securing Sweden's Digital Assets: A retrospective analysis of Database Security in Sweden, with insights from top security experts.

AUTHORS: Hafström and Tomic Lindvall

PUBLISHER: Department of Informatics, Lund University

EXAMINATOR: Osama Mansour, Docent

PRESENTED: May, 2023

DOCUMENT TYPE: Bachelor Thesis

NUMBER OF PAGES: 127

KEY WORDS: Database Security in Sweden, Database Threats, History of Database Security, Database Security Best Practices

ABSTRACT:
This thesis addresses the gap in literature and research surrounding the historical context of database security in Sweden. Through extensive research and in-depth interviews with Swedish cybersecurity experts, it highlights the underrepresented "story" of database security and emphasizes the need for a more holistic approach to this highly specialized field. Key findings indicate that the economic and social costs of security breaches are escalating, which intensifies the necessity for security solutions. The industry's trend towards specialization underscores the need for "generalized specialists" who can provide a broad perspective.

The thesis ultimately argues that a broader perspective is required to better equip society in preempting future cybercrimes. Despite significant advancements in the field, the study identifies a need for historical context and a more integrated approach to database security in Sweden.

Acknowledgments

We would like to express our deepest gratitude to all the experts who found our thesis interesting enough to share their decades of accumulated experiences with us in the hopes of helping secure the future of Sweden's digital landscape. We especially want to thank Christoffer Jerkeby for sharing his network with us and steering us towards people with experience that is strongly and specifically related to our research questions.

Last but not least we want to thank both of our partners who recognized how hectic this time period was for us in conducting this research in unison with beginning our professional careers, and with that stepped up and took extra care of our little families. We'd like to dedicate this thesis to Fanny and Sonja.

Table of contents

Figures

Tables

1 Introduction

1.1 Background

Cybercrime and resultant database security breaches places significant burdens on commercial, economic, reputational, political, and other societal components. It is estimated that the cybercrime economic damage alone in 2020 cost the global economy just under 1 trillion USD (Cremer et al., 2022), or about 1% of global GDP. If we scope down from cybercrime to specifically data breaches, IBM reports an average cost of 392 million USD per incident when dealing with very large breaches (IBM Security, 2022). The average cost of a data breach is climbing, and 83% of the 550 organizations analyzed by IBM have experienced one or more data breaches (IBM Security, 2022). The damages from a data breach can come in many different forms such as reputational damage and financial loss (Buckman, Hashim, Woutersen & Bockstedt, 2019). In extreme cases it can even lead to threats to national security (Wired, 2016). Therefore, securing data, and where data "sits", that is, in databases, should be a critical process. However, it appears that while much has been accomplished around database security there is scant information about database security from a holistic and historical view that can be used to protect data even better in the future.

To the best of our knowledge, a historical overview regarding database security in Sweden does not exist (see: 3.6 Approach to literature selection) and historical overviews of database security internationally seems to be limited and not well researched. The academic literature, the few sources found, has attempted to observe a historical perspective when it comes to database security, mainly in the time period 1980-2008 (Lesov, 2008). Lesov concluded that the "intense focus" on database security in the 30-year time period analyzed "created some welcome improvement", but whatever improvement was made was still outpaced by the threat growth (Lesov, 2008).

On the other hand, industry and state-affiliated entities are more active when it comes to researching, suggesting or regulating how organizations are supposed to implement safe data protection, some examples include the National Institute of Standards and Technology (NIST), the General Data Protection Regulation (GDPR), and the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC27001) (NIST, 2023; ISO/IEC 27001, 2022; GDPR-Info.eu, 2023). The literature suggests that, even while set out to comply with frameworks and regulations, not all organizations are able to achieve success. Organizations may not be entirely ready to ensure that cybersecurity initiatives are properly governed in terms of staff involvement and performance (Kajtazi, Cavusoglu, Benbasat & Haftor, 2018). However, there could also be a lack of engagement on an individual level, which could be due to various elements like cultural, motivational, learning inclinations, and other behavior-related concepts (Chowdhury, Katsikas & Gkioulos, 2022).

A considerable amount of research that is done in the realm of information systems often revolves around an organizational context (Avgerou, 2000). Avgerou (2000), albeit with the risk of oversimplifying things, identifies five primary subject domains within the realm of information systems research. One of these domains is how we can utilize information technology

to facilitate the operations of an organization, and this includes areas of application such as database technologies (Avgerou, 2000; Gregor, 2006).

1.2 Problem statement

Data is essential in the modern world for determining the success or failure of an organization because most information systems rely on databases to store critical information (Ingole et al., 2023). Despite the importance of security, research on cyber risk in the academic field is limited, which could be due to historical data in the field also being limited (Biener et al. 2015). Biener et al (2015), however, also admits that analysis of a historical context in cyber risk could be misleading due to the risk of substantive change in the field. Nevertheless, the research community in the field of databases has been shown to contemplate many concerns well in advance of them being tackled by actual implementations (Lesov, 2008). There has therefore, historically speaking, been a gap or a delay between knowledge in the research community and organizational implementation.

When analyzing security phenomena such as meeting success factors, historical perspectives, and cybercrime, studying and reporting context can also help us better convey the applications of said analysis (Johns, 2006). Context helps illuminate a phenomenon by using its associated surroundings, and some important dimensions of context include: who? where? when? and why? (Johns, 2006). In this regard, it would be particularly interesting to contextualize database security, looking at past experiences and challenges that Swedish experts in the field have faced could provide us with valuable insights for present and upcoming security initiatives.

Sweden ranks among the best countries in the world when it comes to the lowest cost of a data breach (IBM Security, 2022). However, using the existing literature, we cannot pinpoint how or why this is the case. We found very limited contextualized research made regarding a historical perspective, mostly the research made by Lesov (2008). We found no research at all that applied both a historical perspective and a perspective geographically based on Sweden. There is a lack of multilevel research, which results in there not being a tool readily available to analyze the how's and the whys of database security and data breach cost in Sweden.

1.3 Research question

By describing the past twenty years of database security in Sweden, what insights can be gained in the following areas?

- Threats
- Solutions
- Policies
- Future Efforts

1.4 Purpose

The purpose of this thesis is to articulate the past 20 years of database security in Sweden for practitioners in various IT-roles in Sweden. IT-artifacts and related technologies and phenomena will be explained with added layers of higher context in the form of a historical and a Swedish perspective. The description is meant to give a general overview of experiences, challenges and lessons amassed by top security experts operating in Sweden regarding database security. It is our hope that the perspective and explanations we bring forth can aid in securing digital assets and protect organizations by preventing potential shortcomings in database security, and that our contextual variables can enhance the understanding on the subject.

1.5 Delimitations

This study focuses on the protection of credentials stored in databases with the probable use for authentication and/or authorization in computer systems. We have also chosen to focus on databases designed, created, maintained, or otherwise mainly controlled by Swedish entities, be these entities a Swedish person, company, institution or government agency or persons working with any of the above-mentioned entities.

There are several different types of theory when researching in the field of information systems, such as analysis, explanation, prediction and prescription (Gregor, 2006). The scope of our study aims to explain the overall historical context of database security in Sweden, without in-depth analysis nor prescription.

Our scope of a historical context is from the early 2000s to the spring of 2023. Because of the often-sensitive nature of how organizations design and implement security measures of their databases, we are unable to conduct a large-scale data collection of Swedish database security methods. Instead, we have relied on informants with, hopefully, deep insights and an exposure to many Swedish databases.

2 Theoretical Background

This section of the thesis aims to explain the most relevant aspects of database security, including the context, historical background, policies, as well as threats and mitigations. Our thesis problem statement highlights that database security has not been sufficiently covered in a historical and holistic view, especially in Sweden. Therefore, we feel it is prudent to articulate a comprehensive background on the main points (the "must haves") about database security to help frame the insights we attempt to obtain. Volumes can be written about each subsection, so we have chosen to highlight the main points of view and challenges.

2.1 Why context is important

A considerable amount of research in the field of Information Systems (IS) tends to rely on widely accepted and familiar ideas about technology, which often leads to conceptualizing IT artifacts as being "relatively stable, discrete, independent, and fixed" (Orlikowski & Iacono, 2001). This has led to research in the field often taking the IT artifact for granted or presuming it to be unproblematic (Orlikowski & Iacono, 2001). However, Orlikowski and Iacono (2001) argue that a single conceptualization of technology will not work for all usage contexts, and that the technology artifacts "are at the core of context-specific theorizing in IS research". There are also efforts being made to emphasize the potential for IS researchers to use context and contextual variables in multilevel research (Venkatesh et al., 2023). Venkatesh et al. (2023) recognize context as a crucial factor in the development of theories and comprehension of a particular phenomenon, as well as enhancing the understanding of the subject matter.

When conducting multilevel research or theory development, researchers can incorporate context to identify the details in said high level context that affect the lower-level outcomes and relationships (Venkatesh et al., 2023). This type of approach is not common in IS but is described as "sorely needed" due to the fact that it can help understand "where and how the effects of IT are manifested" (Venkatesh et al., 2023). Griffin (2007) describes the importance of context and how it can influence varying strengths, directions, and base rates in the processes you are researching. He also highlights the importance of time as a contextual variable (Griffin, 2007).

2.2 Why the historical context is important

Bannister (2002) tells us that there is a lot to be learned from history and yet as a form of research, the IS community has largely overlooked historical studies. Bannister 2002 continues to name four possible reasons why historical research in IS has been largely ignored, these are:

- [Historical research] is by nature interpretive and, until relatively recently, interpretive research has been poorly regarded by many researchers (Bannister, 2002, p.5)
- Historical research is not a research technique with which IS students are familiar (Bannister, 2002, p.5)

- It involves research methods which IS researchers find uncongenial. (Bannister, 2002, p.5)
- There is little by way of methodological guidance available within the IS literature. (Bannister, 2002, p.5)

Bannister (2002) states that IS as a research field has been strong in the first two bullet points above but less strong with the last two, with ethics stronger than history by giving an example of two IS ethics journals (Australian Institute of Computer Ethics and Journal of Ethics and Information Technology). Bannister (2002) writes "It is the contention of this paper that there remains a distinct shortage of good IS historical studies of the development of information systems in organizations and of how IS influences and even shapes organizations over the long term" (see: p.1). Bannister (2002) adds to this that the study of history offers us a lens through which we can view our present circumstances. Mitev (2014) writes that we as IS researchers lose sight of some of the difficulties posed as we are seduced by new technologies, and that we in IS can suffer from presentism.

2.3 The role of databases as a component of Information Systems

The Oxford Dictionary defines a database as "a structured set of data held in computer storage and typically accessed or manipulated by means of specialized software" (OED Online, 2023). Oracle, one of the world's leading database management companies (Hall, 2019), defines a database as "an organized collection of structured information, or data, typically stored electronically in a computer system." (Oracle, 2022). Oracle continues to explain that most types of databases in operation today are typically modeled in rows and columns organized into tables and that most databases use Structured Query Language (SQL) for querying and writing data. There are a multitude of types of databases available for designers and developers to choose from when building their data stores. Here is a brief overview of the most common types of databases.

Based On	Database Type	Description
Model	Relational	Based on the relational data model storing data in rows(tuples) and columns(attributes) to form a table(relation). Uses SQL to manipulate data. Examples: MySQL, Oracle and MSSQL (Dancuk, 2021).
	Non-Relational (NoSQL)	Non-tabular database used for storing a wide range of data sets. Comes in a variety of types such as document, key-value, wide-column, and graph. Examples: MongoDB, Redis, Cassandra and Neo4j (Dancuk, 2021).
	Object-oriented	Uses an object oriented data model storing data, adding database functionality to object programming languages (Dancuk, 2021).
Location	Centralized	Stored and managed in a single location with access through a network. Single point of failure but easier to maintain, less fragmentation (Iacob & Moise, 2015).
	Distributed	Located as different nodes in a network and logically related by functional relations to be viewed globally as a single database. Provides availability, reliability, performance, more fragmentation (Ozsu & Valduriez, 2011).

Design	Operational (OLTP)	Online transactional processing (OLTP) uses relational databases for fast simple transactions and rapid querying for everyday use like payments, customer data and order management (Sinha, 2021).
	Analytical (OLAP)	Online analytical processing (OLAP) is optimized for large complex data analysis and reporting. Designed for e.g. data scientists, business analysts, business intelligence (BI)(Sinha, 2021).
Hosted	On-Premises	Infrastructure, software and data reside in-house in an organization with total control over security. Often preferred over cloud hosting dealing with very strict compliance and security (Dancuk, 2021).
	Cloud	Regarding public or hybrid cloud models, often provides database-as-a-service (DaaS) avoiding capEx investments as infrastructure, software and data is hosted by third party providers. Scalable, flexible with low staff and maintenance costs, sometimes unclear regulatory compliance with highly sensitive data (Sinha, 2021).

Table 2.3.1: Comparison of databases.

Connolly and Begg (2015) argue that database systems are the most important development in the field of software engineering and that these systems are now the underlying framework of the IS. IS have a set of components that can be found in every computer-based information system that together interact to produce information (Chesney, Stair & Reynolds, 2017). Illustrated in Figure 2.3.1, Chesney, Stair and Reynolds (2017) write that these six components are hardware, software, databases, procedures, people and telecommunications.

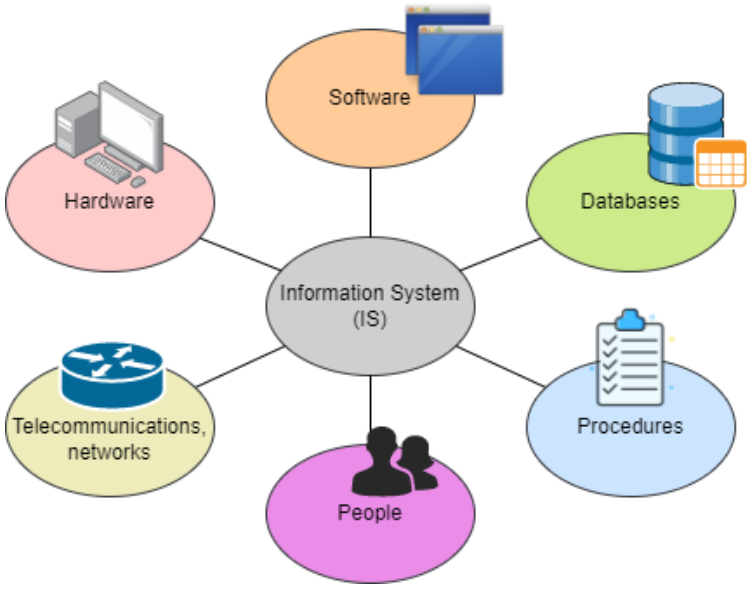

Figure 2.3.1: The six components of computer-based information systems according to Chesney, Stair and Reynolds (2017)

2.4 Historical overview of database security

Lesov (2008), in his study of database security in a historical perspective over a period of thirty years, lays out a summary for each decade what were the main threats as well as what the mainstream research of database security was focused on. In this section we use Lesovs findings, combined with other sources, together with Swedish law enforcement statistics of reported breaches of data security to paint a picture of the past 20 years of database security in Sweden.

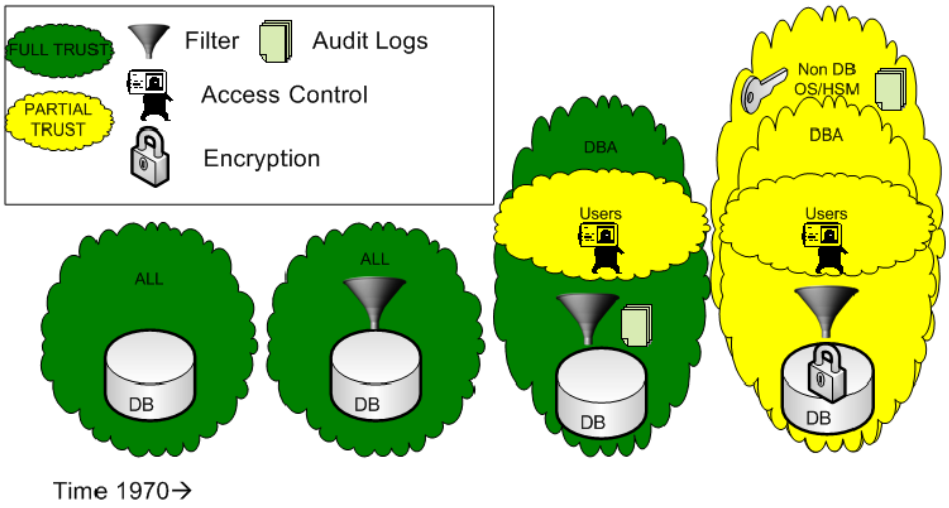

Figure 2.4.1 Database security evolution (Lesov, 2008)

Figure 2.4.1 shows the trend of reported breaches of data security to Swedish law enforcement from 1980 to 2022. Sweden had already in 1973, as the first country in the world, passed legislation aimed at information security breaches, and "hacking" (Collin, 2001). There is believed to be a large number of unreported cases of data breaches to the police (Lindskog *et al.*, 2022). If these are the consequence of not being able to detect intrusions or breaches, unfamiliarity of the law or unwillingness to report is difficult to measure. As an example, the Swedish Police Authority estimates only about 3% of Swedish companies report cybercrime (Söderqvist, 2021). Until 2020 there was no differentiation in reported crimes and included in the same crime code were, DDoD/DoS attacks, malware for extortion purposes, attacks through social media or e-services as well as "other" (Brottsförebyggande rådet, n.d.). Breach of data security (swedish: dataintrång) is defined in the Swedish criminal code as:

> *A person who unlawfully obtains access to information intended for automatic processing, or unlawfully alters, erases, blocks or, in a register, inserts such information, is guilty of breach of data security ... The same applies to a person who seriously disturbs or impedes the use of such information in an unlawful way through some other similar measure (Government.se, n.d.).*

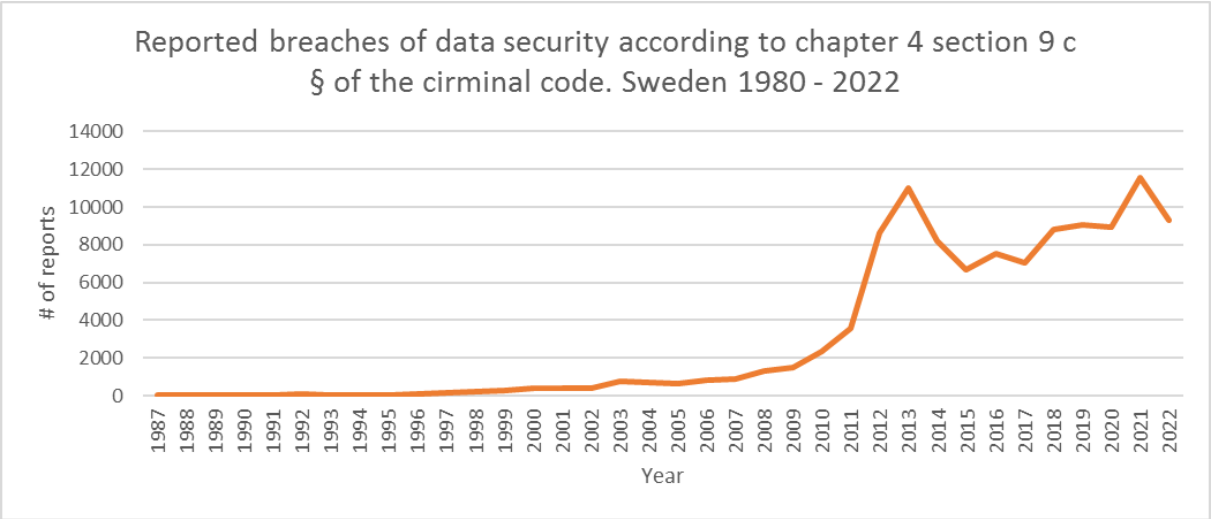

Figure 2.4.2: Reported breaches of data security to the police 1980-1922 (Brottsföreby-ggande rådet, n.d)

2.4.1 The 1990s

Even though our scope for this thesis extends from 2000 until today we have chosen to include background trends, technological and threat advancements from the 90s, as these advancements lay the foundation of much of what happens later.

With the 90s came the advent of the world wide web, personal computers, and an explosion of commercialization. With corporations wanting a presence on the web to reap the benefits, new perimeter defenses in the form of firewalls with rules that dictated what connections from outside the internal network could connect and through which protocols were implemented (Lesov, 2008). During this time users became separated from the databases with frontends, no longer needing SQL knowledge or technical know-how (Lesov, 2008). With firewalls providing network segmentation and filtering, limiting many direct database attacks, the application front-end to the database did not provide input validation leading to SQL-injections (see: 2.7.5 SQL/NoSQL injection attacks) (Lesov, 2008). According to Lesov, database vendors during this time also set database configurations with default credentials letting threat actors compromise databases (see: 2.7.3 Configuration vulnerabilities). He continues that because of the increasing number of users with access to databases during this time period, insider threats also grew (see: 2.7.1 Insider threats) as well as more attack vectors against databases because of increasing complexity and modularity. Lesov also states that this era saw the emergence of HIPAA as more privacy protections were being drawn up.

During this period most research was focused on continuing to refine access controls and encryption, as well as the new up and coming field of privacy protection (Lesov, 2008). Lesov continues to describe a new access control model, Role Based Access Control (RBAC), that was being researched. RBAC came about because of practitioner dissatisfaction with the then dominant Discretionary Access Controls (DAC) and Mandatory Access Control (MAC) models, inspiring researchers to find an alternative, RBAC became the dominant form of access control and is still being used (Jin, Krishnan & Sandhu, 2012). RBAC was adopted for it being able to simplify the task of access control administration, supporting function-based access as well reducing complexity and therefore costs in large scale applications with large scale DBMS vendors like Oracle and Sybase adopting it (Bertino, 2003).

Encryption of databases also became more feasible around the year 2000 with increasing processing power of systems allowing for greater data security, with the possibility of storing the encryption keys externally preventing the database administrator (DBA) or whoever had those permissions to read data in plaintext (Lesov, 2008). Although there were teething problems in that optimization of encrypted databases was poor, with the entire database needing to be scanned if even one table was encrypted to provide query results (Lesov, 2008). Still used at this time were the Data Encryption Standard (DES), first published in 1975, and 3DES (applies the DES algorithm three times to each data block) was first published in 1981 (Vmerkle & Hellman, 1981). In 1997 the National Institute of Technology (NIST) announced the Advanced Encryption Standard (AES) initiative to find a replacement of the aging DES algorithm, with work on the standard continuing into the early 2000s (Gilchrist, 2003).

Already in the late 1980s there was research conducted on how to detect anomalies and real time intrusions in computer systems, with Dennings research setting the foundation for Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) (Denning, 1987). In the early 90s researchers developed real time intrusion detection systems that allowed review of ongoing attacks, allowing real-time response (Kemmerer & Vigna, 2002). In the late 90s host-based IDS systems were researched and products released on the market such as Snort (both a network and host IDS tool) which allowed for IDS systems to move from network monitoring to host monitoring, as networks became faster, analyzing packets on hosts became more feasible (Bruneau, 2001).

During this decade, reporting of data security breaches in Sweden went from the low dozens annually to escalating to several hundreds per year in the latter half of the decade (Brottsförebyggande rådet, n.d.) (Figure 2.4.3).

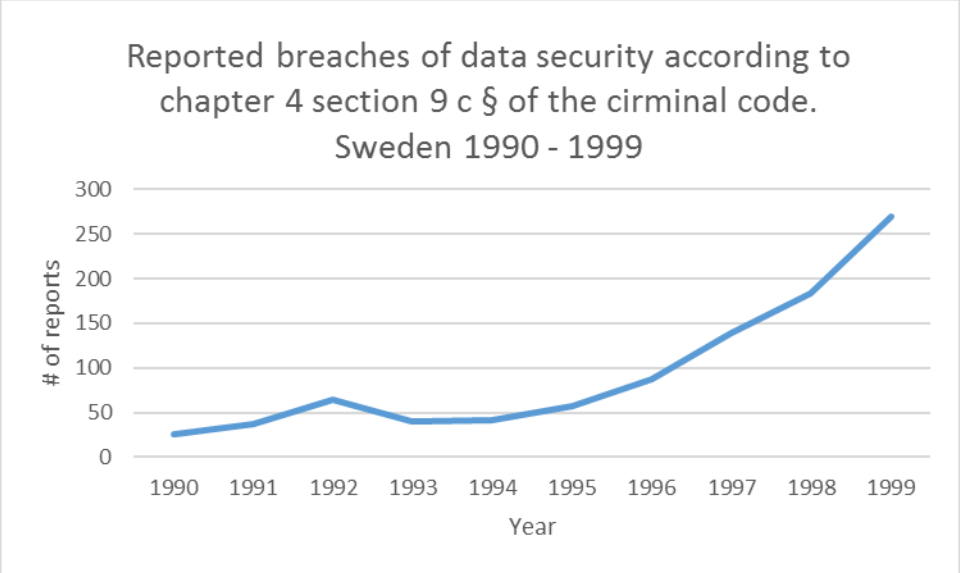

Figure 2.4.3: Reported breaches of data security to the police 1990-1999 (Brottsförebyggande rådet, n.d.)

It was also during the 90s that saw the development and publications of cryptographic hash functions often used to secure credentials in databases (password verification by hash comparison), i.e., storing credentials such as passwords as hashed versions to be later compared to the

provided hash of the user to allow login to a service (Preneel, 2010). Notable hash functions published during this decade are:

- MD5, published 1992 (Rivest, 1992)
- SHA-1, published 1995 (National Bureau of Standards, 1995)
- RIPEMD-160, published 1996 (Dobbertin, Bosselaers & Preneel, 1996)
- bcrypt, published 1999 and based on Blowfish (Provos & Mazieres, 1999)

2.4.2 2000 to 2010

With the emergence of Web 2.0 and social networking platforms, personal information became more accessible, and databases kept growing with information security attacks becoming a significant business risk, with attackers using new tools, trying to minimize detection and profiting (Lesov, 2008). Security researchers around databases during this era saw new fields emerge, like privacy protection, not driven on inference attacks but how to anonymize private data shared with third parties (Lesov, 2008).

There was also a new addition to access control models in the early 00s. Attribute-Based Access Control (ABAC) came as an alternative to RBAC and differed in being attribute based, including user attributes, environmental attributes, and resource attributes to control access (Shu, Yang & Arenas, 2009). ABACs contextual attributes can be, for example, time of day, geographical location and evaluated threat level (Owasp, n.d.).

In the early 2000s with SQL injection threats starting to become a threat to databases, firewalls that filtered with port/protocol rules were no longer sufficient and IDS started to be adopted looking for exploit signatures but not vulnerability signatures, for any given vulnerability there can be 100 different ways to exploit it (Pirc, 2017). Even though Intrusion Prevention Systems (IPS), systems that not only could alert malicious behavior but also block it, had now been developed, adoption was low as organizations feared it could block harmless traffic (Pirc, 2017). It was first in the latter half of the 2000s that organizations started to adopt both IPS and combined IDS/IPS systems that now offered countermeasures such as pattern matching, string matching, anomaly detection, heuristic based detection as well as blocking of known command and control IP addresses (Pirc, 2017).

Following the trend of the 90s, reported breaches of data security in Sweden continued to rise during this decade, starting at a few hundred a year to over 1500 a year in 2009 (Brottsförebyggande rådet, n.d.) (Figure 2.4.4).

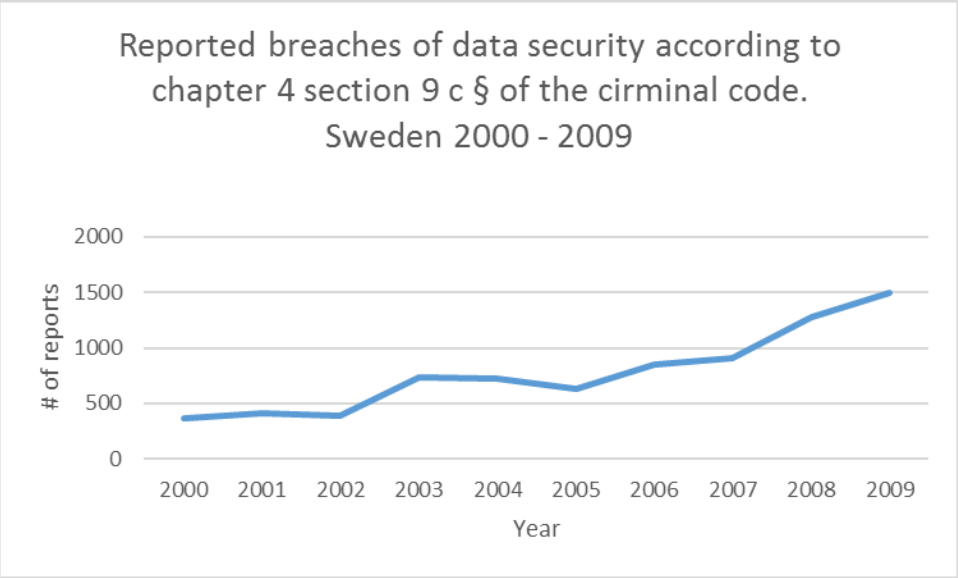

Figure 2.4.4: Reported breaches of data security to the police 2000-2009 (Brottsförebyggande rådet, n.d.)

Database security research during this decade was also starting to focus on attack detection, with attackers attempting to hide their intrusions researchers had to tackle the task of detecting a compromise and products for this task were developed (Lesov, 2008).

Notable hash functions published during this decade are:

- PBKDF2, published 2000 (Kaliski, 2000)
- SHA-2 family: SHA-256, SHA-384, SHA-512, published 2002 (Preneel, 2010)
- SHA-3 family, published 2012 (Pfautsch *et al.*, 2020)

For a long time, homomorphic encryption, an encryption scheme that allows data to be processed and analyzed as ciphertext without the need for decrypting it first, was theoretical. It was not until 2009 when Dr. Craig Gentry published his dissertation that explained how homomorphic encryption could be implemented in real world applications (Gentry, 2009).

2.4.3 2010 to 2023

Even though the concept of zero trust was first coined by security researcher Stephen Paul Marsh in 1994 (Marsh, 1994) it took several years until the ideas about zero trust started to gain traction. Notable milestones being Forresters Zero Trust model of information security published in 2010 by analyst John Kindervag (Kindervag, 2010), creating a model for a zero-trust network as well as OSSTMM (Open Source Security Testing Methodology Manual) first referencing it in the early 2000s to eventually providing a whole chapter about zero trust in in 2010 (Herzog, 2010). Google revealed in early 2010 that they, as well as several other large US tech companies, had been targeted by Chinese sponsored hackers, feeling that the old way of perimeter security was inadequate Google started working on a new approach to access management called BeyondCorp, used internally starting in 2011, that was an implementation of the zero-trust model (Vergadia, 2022).

Google published a lot of their findings and practices on how to make zero trust work (Ward & Beyer, 2014) but adoption was still not rapid, and it took a few more years for standards

organizations and regulatory bodies to start to recommend zero trust to a broader audience. With NIST starting work on a special publication in 2018 (NIST, n.d.) that was released in 2020 entitled "SP 800-207, Zero Trust Architecture" (Rose et al., 2020) and the UK National Cyber Security Centre (NCSC) recommending it as security guideline in 2020 (National Cyber Security Centre, 2021).

Microsoft reported in their Zero Trust Adoption Report for 2021, that 96% of 1200 security decision makers that responded said that zero trust was critical for their organization's success (Microsoft Security, 2021). By using ABAC, providers could have enough information from users to provide conditional access depending on a range of factors (Microsoft 365, 2019)

The idea and use of multi factor authentication (MFA) has been around for some time outside of mainstream IT-services such as in the banking sector with pin numbers required to access ATM machines started as early as 1967 (Milligan, 2007). During the late 80s the RSA company started selling keyfobs displaying a rotating series of characters to be used as two-factor authentication (2FA) for government and companies dealing with sensitive information. Not much adoption for 2FA was seen at IT and online service companies the following decades until 2010 when Google, with others, were targeted by Chinese state sponsored hackers (Bankston, Schulman & Woolery, n.d.), with Google taking public steps to implement new security features, one of them was releasing the Google Authenticator phone application the same year for iOS and Android. This app provided the similar rotation series of six-digit codes but could now be used in any phone to authenticate together with other credentials against Google, "something you have and something you know", and soon many other services that used the standard (Bankston, Schulman & Woolery, n.d.).

A year later Google made it an option to enable 2FA on all their accounts. In the coming three years Facebook, Yahoo, DropBox, Twitter and Microsoft started rolling out 2FA for their accounts and a very public iCloud photo leak was publicized, where threat actors had access celebrites accounts and shared their intimate photos, leading Apple to also enable 2FA on their iCloud backups (Bankston, Schulman & Woolery, n.d.). In 2014 Universal Second Factor (U2F) was introduced and let users authenticate not only by an app in a phone but through a USB, and later bluetooth device or biometrics (UAF, FIDO2) (FIDO Alliance, 2018). In the coming years 2FA and MFA become more and more adopted, securing everything from users' social media accounts to administrators' access to systems. New approaches are also being developed during this decade with Google as an example looking at metrics on typing patterns, location information, behavior data etc to allow for conditional access (Bankston, Schulman & Woolery, n.d.).

Since Gentrys publication on homomorphic encryption in 2009, the technology has gone from academic research to applicable use (IEEE, n.d.). The early 2010s new generations of full homomorphic encryption (FHE) were developed, improving on Gentrys blueprint with more and more efficient design, as well as some vulnerabilities found and addressed, and starting around 2020 more practical applications are being seen developed (van den Nieuwenhoff, 2021). Target areas for using FHE are highly sensitive databases such as medical databases, voting records and genomic research (IEEE, n.d.).

The 2010s saw an advancement of IDS/IPS systems that now could now perform better deep packet inspection as encrypted traffic grew as well as addressing application control and phishing countermeasures such as detecting and removing suspicious files before phishing victims could open them (Pirc, 2017). These newer systems on the host level could now better look at database behavior and alert/block suspicious activity.

Between 2010 and 2012 a mainframe belonging to Logica (now CGI) was compromised and databases belonging to the Swedish Tax Authority (sv. Skatteverket), the Swedish state personal address register (sv. Statens personadressregister, SPAR) and the Swedish Enforcement Authority (sv. Kronofogdemyndigheten) were exfiltrated (Ryberg, 2013). A list of vehicles used by the police was obtained as well as some 10,000 social security numbers belonging to people with protected identities were shared online. The Pirate Bay-founder Gottfrid Svartholm Warg (a.k.a. Anakata) was convicted and sentenced to prison for these crimes (Svea Hovrätt, Dom B 6402-13).

In 2015 Sweden's Transport Agency decided to outsource the management of its databases and other IT services to IBM offices in Romania, Serbia, and the Czech Republic with employees in these offices being able to access the data without any security clearances. These databases included every vehicle in the country including police and military vehicles, as well as details of persons in witness protection programs and individuals in the military special forces, e.g., their driver ID photos, with secret identities exposed (Chirgwin, 2017; Mårtensson et al., 2017). The agency in 2016, claiming user error, emailed out the entire database in cleartext to marketers, both foreign and domestic companies, who were subscribers, later discovering what they had done they asked the marketers to remove secret identities by sending out a list which entries had secret identities (Lindhe, 2017).

As with the trend of the 90s and 00s, reported breaches of data security continued to climb with a peak around 2013 (11,101 reported cases) and then again in 2021 with the highest numbers ever reported (11,566 reported cases) (Brottsförebyggande rådet, n.d.). Compared with the start of the millennium, the year 2000 saw 375 cases reported (Brottsförebyggande rådet, n.d.).

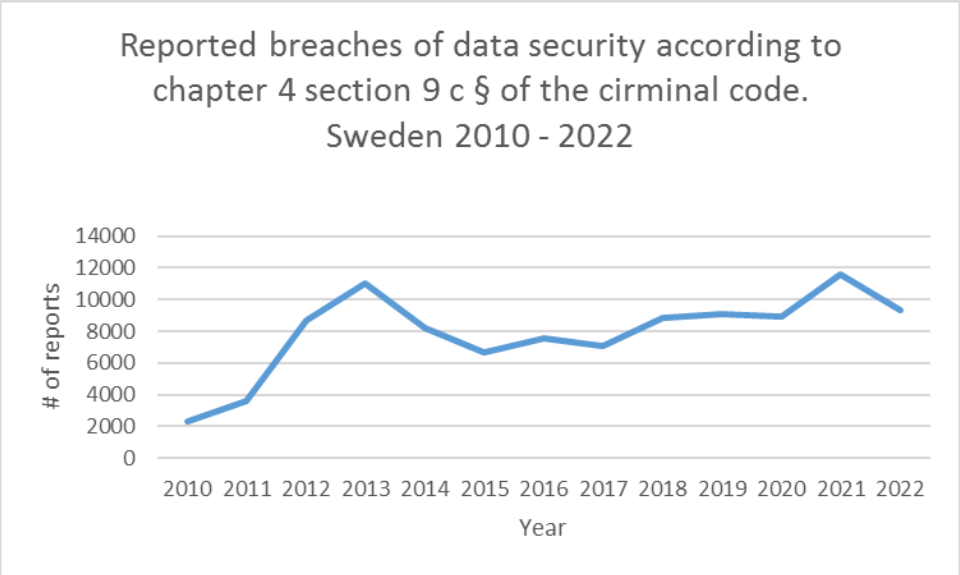

Figure 2.4.5: Reported breaches of data security to the police 2010-2022 (Brottsförebyggande rådet, n.d.)

Notable hashes published during this decade are:

- PBKDF2, the newer version with published 2017 with more pseudorandom functions and is recommended for hashing (Moriarty et al. 2017)

- Argon2, the winner of the Password Hashing Competition in 2015 comes in two three flavors: Argon2d, maximizes resistance to GPU cracking, Argon2i, optimized to resist side-channel attacks and Argon2id is a hybrid version (Wetzels, 2016).

2.5 Confidentiality, Integrity & Availability

2.5.1 The CIA Triad

The CIA triad (confidentiality, integrity, and availability) serves as a foundational model and industry standard for information security (Van Der Ham, 2021; Lundgren & Möller, 2017; von Solms & van Niekerk, 2013). The main principle of the triad is that the security of data as an asset is defined by three aspects: confidentiality, integrity, and availability (Van Der Ham, 2021; Lundgren & Möller, 2017; von Solms & van Niekerk, 2013).

1. Confidentiality, which refers to the protection of sensitive data from unauthorized access and disclosure.
2. Integrity, which involves ensuring the accuracy, consistency, and trustworthiness of data throughout its lifecycle.
3. Availability, which relates to the assurance that data and services are accessible to authorized users whenever needed.

(Van Der Ham, 2021; Lundgren & Möller, 2017; von Solms & van Niekerk, 2013).

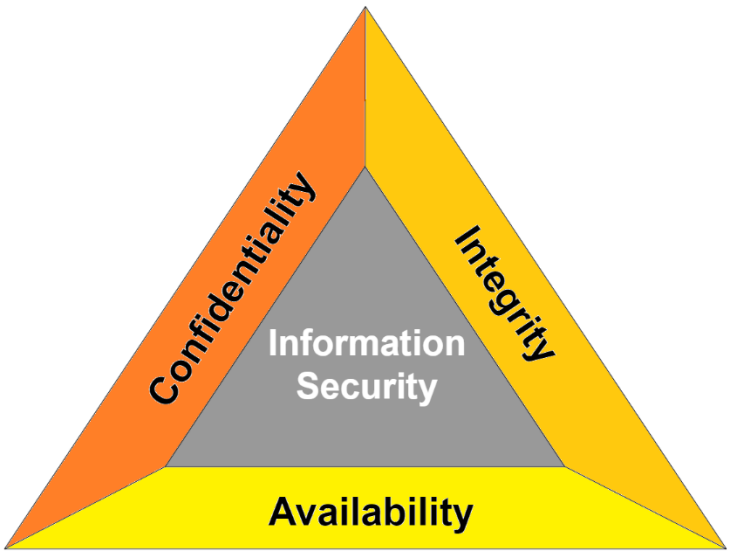

Figure 2.5.1: The classical representation of the CIA-triad.

2.5.2 CIA in database security

Elmasri and Navathe (2015) outline three threats to databases in the context of the CIA-triad as seen below:

> **Loss of confidentiality.** Database confidentiality refers to the protection of data from unauthorized disclosure. The impact of unauthorized disclosure of confidential information can range from violation of the Data Privacy Act to the jeopardization of national security. Unauthorized, unanticipated, or unintentional disclosure could result in loss of public confidence, embarrassment, or legal action against the organization.
> (Elmasri & Navathe, 2015, p.1123)

> **Loss of integrity.** Database integrity refers to the requirement that information be protected from improper modification. Modification of data includes creating, inserting, and updating data; changing the status of data; and deleting data. Integrity is lost if unauthorized changes are made to the data by either intentional or accidental acts. If the loss of system or data integrity is not corrected, continued use of the contaminated system or corrupted data could result in inaccuracy, fraud, or erroneous decisions. (Elmasri & Navathe, 2015, p.1123)

> **Loss of availability.** Database availability refers to making objects available to a human user or a program who/which has a legitimate right to those data objects. Loss of availability occurs when the user or program cannot access these objects.
> (Elmasri & Navathe, 2015, p.1123)

Khalaf (2017) and (Elmasri & Navathe, 2015) list four major control measures to protect the core database, these consist of Access Control, Inference Control, Flow Control and Data Encryption. These are shown in Figure 2.5.2.

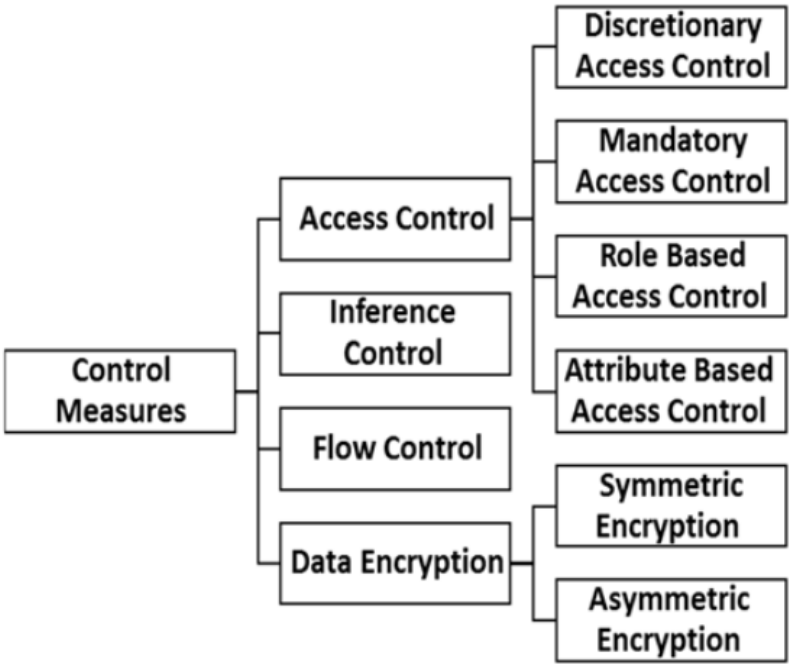

Figure 2.5.2: Control measures (Khalaf, 2017).

Access control is a mechanism used to ensure confidentiality of data in a database by inspecting user rights to access data against a set of permissions (Khalaf, 2017). This has historically been done by MAC, DAC, RBAC and ABAC.

Factors	DAC	MAC	RBAC	ABAC
Access Control to Information	Through owner of data	Through fixed rules	Through roles	Through attributes
Access Control Based on	Discretion of owner of data	Classification of users and data	Classification of roles	Evaluation of attributes
Flexibility for Accessing Information	High	Low	High	Very high
Access Revocation Complexity	Very complex	Very easy	Very easy	Very easy
Support for Multilevel Database System	No	Yes	Yes	Yes
Used in	Initial Unix system	The U.S. department of defense	ATLAS experiment in CERN	The Federal government

Figure 2.5.3: Comparison between DAC, MAC, RBAC, and ABAC (Khalaf, 2017)

Inference control prevents users from inferring confidential data/information though reading summary or statistical information from the database (Khalaf, 2017).

Flow control prevents information from flowing in such a manner that it reaches unauthorized users, and an example of flow control is a system preventing a service application from leaking customer confidential data by specifying channels which information is allowed to move (Elmasri & Navathe, 2015).

Encryption is the last control measure listed and is defined as the conversion of data into ciphertext that cannot be easily converted back into readable information by an unauthorized actor (Elmasri & Navathe, 2015). Elmasri and Navathe (2015) state that the employment of encryption provides an additional layer of security and privacy when access controls are bypassed. Figures 2.5.4 and 2.5.5 show the difference between a table's contents in plain text and the encrypted form (Singh & Kaur, 2015).

Username	Email_ID	Password
Raman	ramanarora@yahoo.com	coolraman
Ravi	raviparkash@gmail.com	123ravi123
Parneet	Parneet22@yahoo.com	pari123parneet

Figure 2.5.4: Database in Normal Form (Singh & Kaur, 2015)

Username	Email_ID	Password
&@*@%	&@*@%@&$&@@y@h$$>$*	>$$l&@*@%
&@?	&@?=@&k@sh@g*@l>$*	ASD&@?ASD
=@&%}}!	=@&%}}!22@y@h$$>$*	=@&ASD=@&%}}!

Figure 2.5.5: Database in encrypted form (Singh & Kaur, 2015)

2.5.3 Criticism of the CIA-triad

Some research in the field of information security criticizes the CIA triad for its overly technical concentration and scope, saying that it is less useful when broader organizational and societal aspects of security require consideration (Dhillon & Torkzadeh, 2006; Anderson, 2003). However, the research doesn't entirely discard the triad, but rather attempts to add additional key terms to it as to widen its scope and utility (Samonas & Coss, 2014). The most commonly additional concepts when talking about revising the CIA triad are authenticity and accountability (Stallings & Brown, 2018)

- **Authenticity**: the capacity to be confirmed and relied upon are essential; this involves having faith in the legitimacy of a conveyed message or its originator. This implies the necessity to authenticate the identities of users and ensure that every input processed by the system is derived from a dependable source. (Stallings & Brown, 2018)

- **Accountability**: necessitates that an action of an entity can be unequivocally attributed and traced back to that entity. This supports post-incident recovery and legal proceedings, aids in detecting and preventing intrusions, and supports deterrence and nonrepudiation. Given the reality that fully secure systems are beyond our reach, the capability to pinpoint a responsible party in the event of an incident is crucial. The concept of accountability also requires systems to maintain logs of operations, enabling future forensic investigations to trace breaches. (Stallings & Brown, 2018)

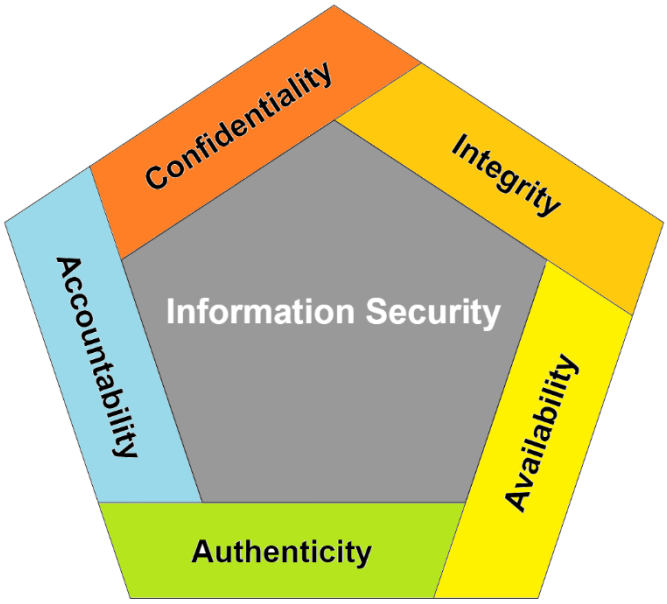

Figure 2.5.6: Variant of the CIA-triad

2.6 Existing Policies and Regulations

In this section we provide an overview and historical context of three policy and regulatory frameworks used by organizations to protect their data.

2.6.1 ISO/IEC 27000

The ISO/IEC 27000 series, sometimes referred to ISO27K for short, comprises more than 20 documents regarding information security standards and is published jointly by ISO and the IEC (Santos, 2021). Santos (2021) describes shortly the first six documents in the ISO27K series as providing guidance and recommendations for "establishing, implementing, operating, monitoring, reviewing, maintaining, and improving an Information Security Management System". The ISO/IEC 27000 series can be applied within all organizations, regardless of size or type of business (Swedish Institute for Standards, SIS, n.d.-a). Figure 2.6.1 shows the series of ISO 27000 with a selection of the most relevant standards regarding cyber, data and information security for organizations according to the SIS.

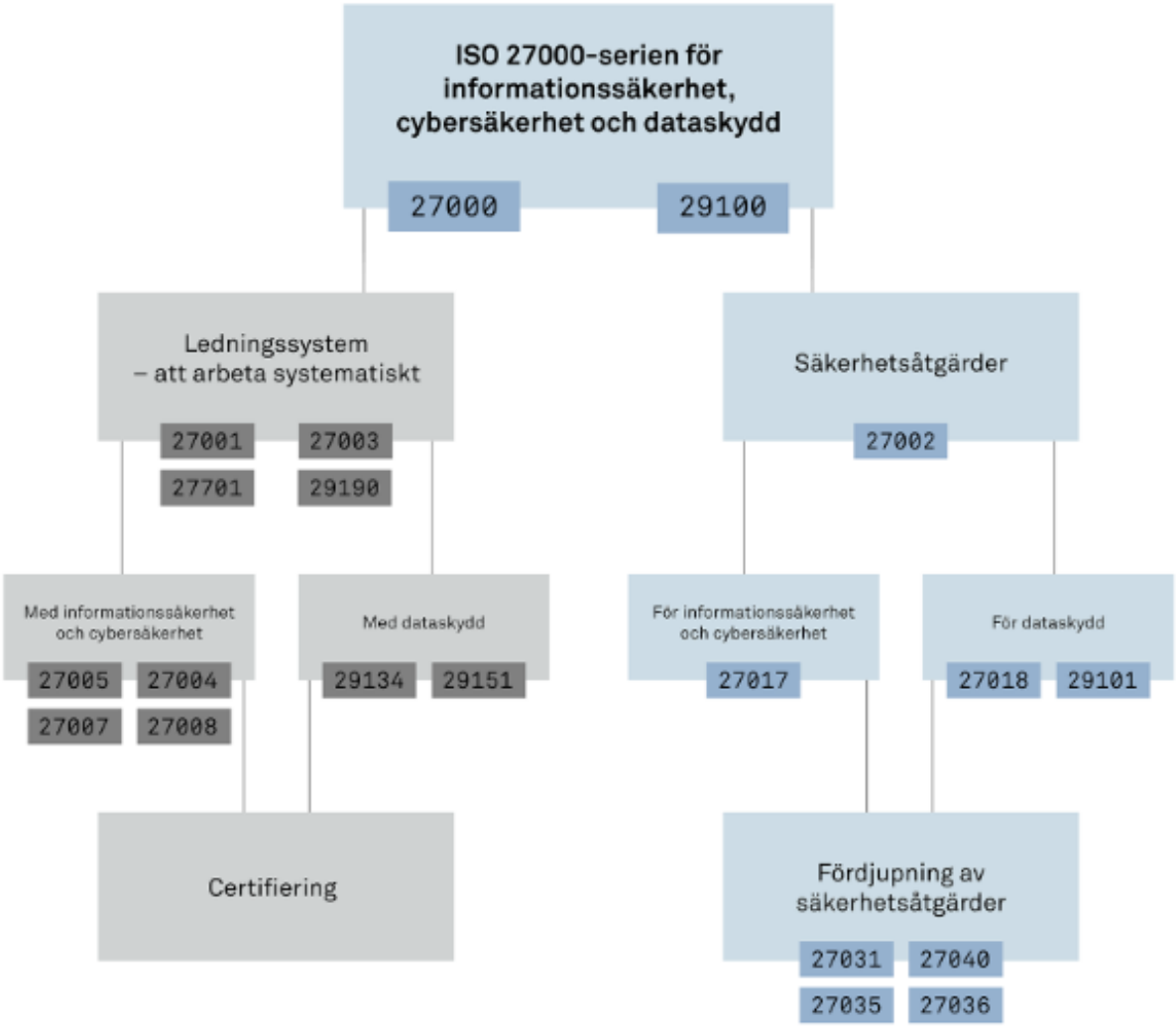

Figure 2.6.1: Tree of ISO 27000 (Svenska institutet för standarder, SIS, n.d.-d)

As seen in Figure 2.6.2 by Mirtsch, Kinne and Blind (2020) when comparing diffusion of three common ISO management system standards from the year they became certifiable, the ISO 27001 certification has by far the largest growth rate but overall low adoption.

ISO 9001 is a quality management system (ISO, n.d-b.) is commonly used by many services and goods producers to help provide products and services that meet customer and regulatory requirements and in Sweden, a new company is certified according to ISO 9001 every day (Svenska institutet för standarder, SIS, n.d.-b). ISO 14001 is an environmental management system (ISO, n.d.-a) and is used by organizations to increase efficiency in the use of resources and materials per delivered benefit (Svenska institutet för standarder, SIS, n.d.-b).

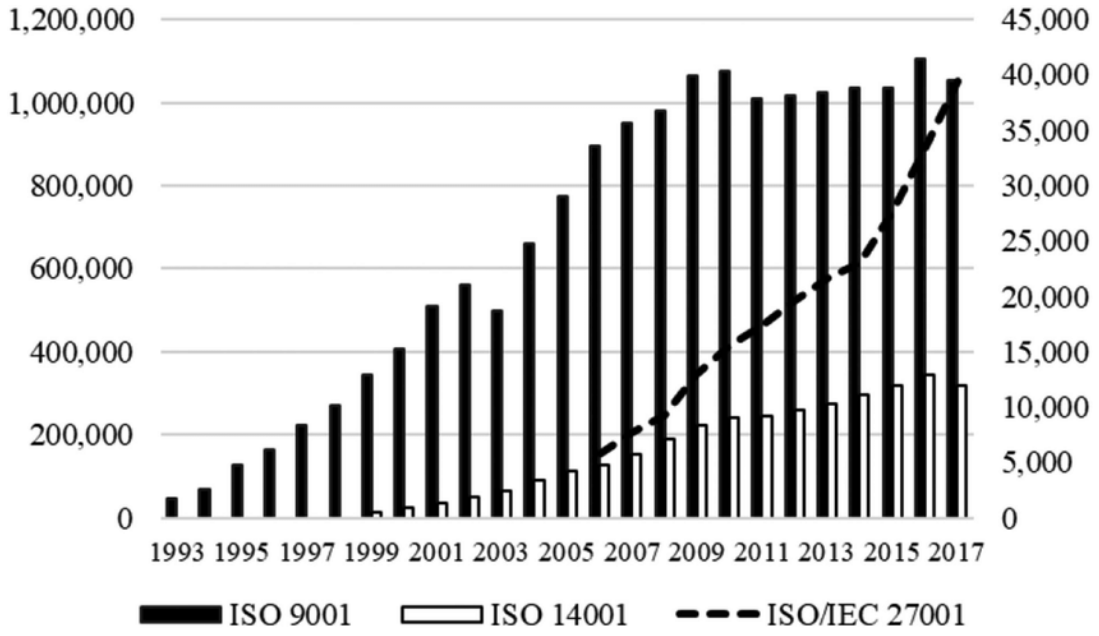

Figure 2.6.2: Evolution of ISO 9001, ISO 14001, and ISO/IEC 27001 over time in terms of valid certificates worldwide (Mirtsch, Kinne & Blind, 2020)

Mirtsch, Kinne and Blind (2020) state that it is expected that ISO/IEC 27001 adoption will rise as firms increasingly store their information in digital information systems and governments and suppliers more and more require firms to ensure information security.

Shown in figure 2.6.3 and according to Disterer (2013), although an international standard, adoption is not evenly distributed and in 2013 Disterer concluded that the large Asian adoption could be explained by Asian based companies offering IT services outsourcing largely to U.S. and European firms. Disterer (2013) believes that the low number of North American adoptees is a confirmation of a common assumption that international IT-standards do not draw much attention there, Disterer continues to state that the standard is widely disseminated in Europe.

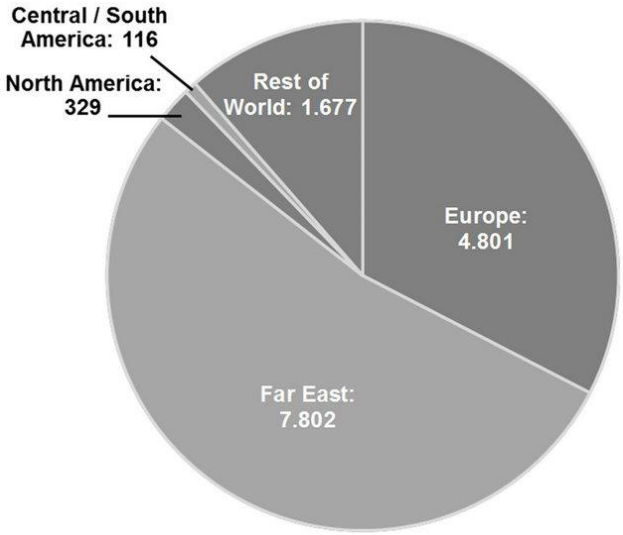

Figure 2.6.3: Number of certificates accord. ISO 27001 by regions (Disterer, 2013)

MSB (n.d.) states in their information portal, "Information security, cyber security and secure communications", that application of the Swedish and international standards SS-ISO/IEC 27000 series facilitates work with information security inside organizations and improves the possibility of externally assessing security and revise it in a uniform way.

2.6.2 GDPR

The General Data Protection Regulation (GDPR) represents "the most significant change to European privacy laws in the last two decades." (Cusick, 2018). GDPR aims to regulate how businesses manage the data of EU citizens (MSP360, 2018). All businesses and organizations that not only operate within the EU, but store any type of data about EU citizens, will need to comply (Mansfield-Devine, 2017). The intention behind it being to protect the fundamental rights and freedoms of people, in particular the right to protection of personal data (GDPR-info.eu, 2023).

The primary focus of the GDPR revolves around ensuring data security. GDPR does not explicitly outline specific data protection measures or precise retention guidelines, but mandates that all essential steps be taken to avert potential data breaches and retain data only for the duration that is legally permissible (MSP360).

"What does 'good' look like? Because the GDPR isn't highly prescriptive in that regard, in terms of defining with any exact detail the nature of security controls" (Mansfield-Devine, 2017). GDPR leaves it up to the organizations how they make their data secure, and if they misjudge their efforts, they risk punishment. The regulation makes it possible for fines up to 4% of global turnover or 20 million euros (Mansfield-Devine, 2017). The idea behind this is to get a holistic view of the security process, instead of a "tick-box compliance exercise", which usually ends up being the case with prescriptive regulations (Mansfield-Devine, 2017).

However, there are a few exceptions to the "non-prescriptiveness", one being encryption. (Mansfield-Devine, 2017). GDPR requires that personal data is always encrypted with "state-of-the-art" measures, meaning both when the data is in rest and in transit (Shah, A. et al, 2019).

2.7 The need for secure databases

The expanding threat landscape against databases is a challenge for any organization relying on such a database. The following section covers an overview of potential consequences of exploited databases according to existing research.

2.7.1 Loss of trust and reputation

The integrity of an organization's databases is crucial to maintaining trust and reputation. According to a report by Forbes Insights in association with IBM (2014), security breaches have the potential to do the most reputational damage to an organization compared to other common threats. Aon's Global Risk Management Survey ranks "Damage to Reputation / Brand" as the number one risk an organization can face overall, as voted by thousands of risk managers across 60 countries and 33 industries (Aon, 2019). Furthermore, they rank cyber-attacks and data breaches as their third highest risk (Aon, 2019). Following a data breach, the damage can also be amplified by negative media coverage (IBM, 2014). The consequences of losing your brand image can have a profound and lasting impact on customer attrition and competitive advantage (IBM, 2014).

2.7.2 Regulatory consequences

With the advent of data protection laws such as the European GDPR leaks and breaches of databases that are not handled according to GDPR regulations can be very costly for the organization. With less severe infringements finable up to €10 million or 2% of the firm's worldwide annual revenue, or if more severe, up to €20 million or 4% of the firm's worldwide annual revenue whichever is higher (Mansfield-Devine, 2017). The fines might also increase more if the database was leaked, and it was later found that sensitive personal data was stored against GDPR regulations according to the different criteria GDPR uses to determine the fines (Wolford, 2018).

2.7.3 Ransomware

A ransomware attack is an attack based on getting or blocking access to an asset, and then holding it hostage while demanding compensation from its rightful owner (Young & Yung, 1996). According to Young and Yung (1996), it is becoming increasingly popular to run this type of attack against organizations, because you give them an "opportunity to mitigate their losses by paying a ransom". Being subjected to a ransomware attack can impact organizations to a point where they can no longer deliver vital services, and the impacts have proven challenging from an economical and reputational perspective for both large and small companies (CISA, n.d.).

2.7.4 Financial losses and costs

Various consequences resulting from data breaches or leaks tend to lead to some kind of financial loss, either indirectly by losing revenue due to loss of reputation and trust, or directly by regulatory causes. The extent of financial loss can vary a lot, but in extreme cases where leaks are particularly sensitive, organizations can be forced to pay settlement fees of hundreds of millions of dollars. It is estimated that cybercrime cost the global economy just under 1 trillion USD world-wide in 2020 (Cremer et al., 2022). For example, Equifax, a consumer credit reporting agency failed to take reasonable steps to secure its network and data in 2017 (Federal Trade Commision, 2021). This caused a data breach and subsequent leak of customer's personal information affecting approximately 150 million people, which ultimately led to a 575-million-dollar settlement to compensate for fraud and identity thefts. In 2022 the average total cost of data breaches per year amounted to staggering sums per year, as seen in billions of dollars in Figure 2.7.1

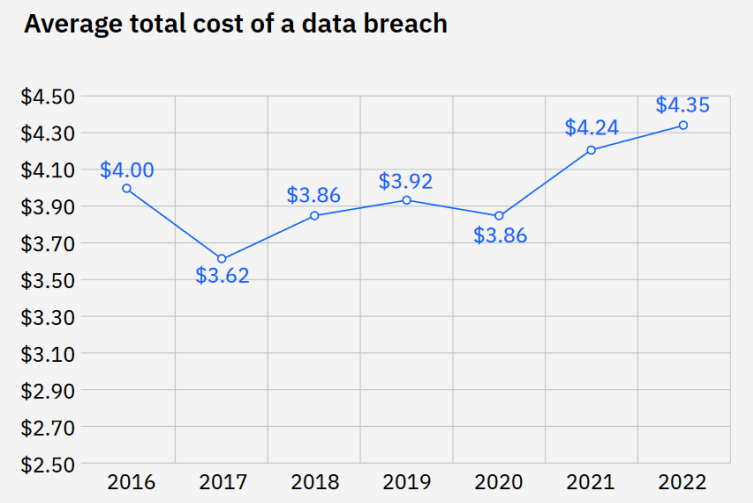

Figure 2.7.1: Average total cost of global data breaches in USD millions (IBM 2022).

2.7.5 National security concerns

There have been several large disclosures of database breaches in recent years leading to national security concerns. One of these disclosures is the Aadhaar data breach in India that might have involved as many as 1.2 billion Indian users. The Aadhaar system was created to cut down on bureaucracy and unify Indian IDs under one system. The Aadhaar system provides Indian citizens a 12-digit unique-identity number and collects biometric information about the citizen such as fingerprints, retina scans as well as face photographs (Perrigo, 2018). It is the biggest biometric database in the world with around 89% of India's population's data stored in the system (Tiwari, 2018). It was reported in 2018 that access to the Aadhaar system could be bought for as little as 500 rupees, or $7.8 (BBC News, 2018).

The U.S. federal government suffered a major database breach when Chinese hackers stole a huge trove of records from the OPM (US Office of Personnel Management) including 18 million copies of 127-page background checks on federal employees, with questions ranging from personal finances, drug use, lie detector results as well as complete personnel files of 4.2

million federal employees and 5.6 million images of government employee fingerprints (Koerner, 2016). The attack is thought to have been executed by first gaining a foothold inside OPMs network with a contractor's credential and then once inside gaining access to the domains users and trying each one until they found one with enough permissions to be able to access a jump box to gain access to the files they sought (Koerner, 2016).

2.7.6 Potential for Credential Stuffing

The wide-spread availability of credentials exposed by data breaches has trivialized criminal access to billions of accounts (Thomas *et al.*, 2019). A Google/Harris poll conducted 2018 on 3000 adults ranging from 16 to 50+ years in age living in the U.S. concluded that 52% of informants reuse the same password on multiple accounts and 13% reuse their password on all accounts (Google, 2019). The consequences of successful access can result in users accounts being compromised and further exploitation such as identity theft, unauthorized purchases, fraud, extortion, facilitation of sending spam and/or malware to the user's contacts. Credential stuffing is a subset of the brute force attack category in that the attacker will try to automate injection of stolen credentials from a database breach into other website login portals to gain access to multiple accounts (Mueller, n.d.).

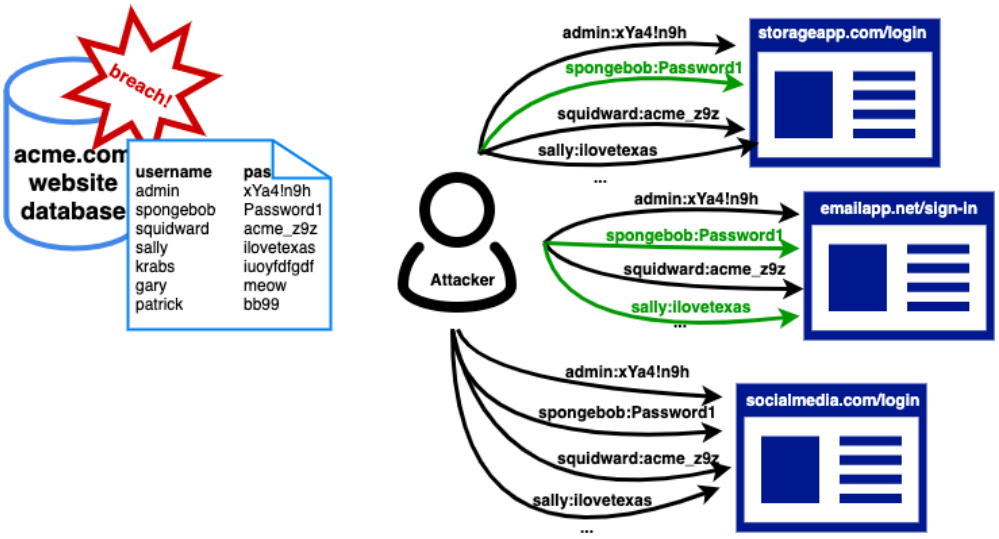

Figure 2.7.2: Credential stuffing (Mueller, n.d.)

2.8 Common threats and challenges

In this section we provide an overview of the most common threats to database security, as well as recommended mitigations according to the literature.

2.8.1 SQL Injection Attacks

Threat

SQL Injection Attacks (SQLIA) are attacks that are often targeted against web applications that use on the fly SQL queries to the backend database without applying proper user input validation (Kindy & Pathan, 2011; Sarmah, 2019). There are many techniques to SQLIA, here is a quick overview of three of them according to Alwan and Younis (2017) and Kindy and Pathan, (2011):

Tautologies based attacks inject conditional SQL statements queries for the backend to evaluate if the condition is true, e.g. (1=1) or (--). Often this is used to bypass authentication on web pages to gain access to the database. Example SQL statement:

```
Select * from student where student_ID='1' or
'1=1'--' AND student_password='1234'
```

Figure 2.8.1: Tautology based SQL injection

Logically incorrect queries are used to generate often useful error messages from the database to facilitate further correct injections. Example SQL statement:

```
SELECT * FROM studentTable WHERE student_id='1111'
AND password='1234' AND CONVERT (char, no)
```

Figure 2.8.2: Incorrect query SQL injection

In union query injections the attacker inserts additional statements into the original query using a UNION query that can lead to the database returning the results of the original query together with the results of the injected query. Example SQL statement:

```
SELECT * FROM studentTable WHERE student_id='1111'
UNION SELECT * FROM memberTable WHERE
member_id='admin'--' AND password='1234'
```

Figure 2.8.3: Union query-based SQL injection

Mitigations

Kindy and Pathan, (2011) recommend the use of mitigations such as:

- Prepared statements: creating a fixed query "template" providing specific placeholders for input data.
- SQL-IDS: SQL Intrusion Detection System that monitors applications for injection attacks in real time.
- Stored procedures: Server side procedure written in SQL accessible by the DB engine that can be called upon by the application instead of relaying user input data.
- Manual approaches: Conduct code reviews and defensive programming.

2.8.2 Misconfigurations

Threat

Databases are often misconfigured or have accounts and configurations set to defaults (Sarmah, 2019; Furmanyuk, Karpinskyy & Borowik, 2007). The occurrence of database default credentials and configurations can lead to exploitation of the database (Mousa, Karabatak & Mustafa, 2020). Mousa, Karabatak and Mustafa (2020) continue to also note the occurrence of databases inaccessible due to miss-configurations.

In December 2016 it was discovered that the NoSQL database MongoDB, ranked as the 5th most popular database management systems in 2022 (*Db-engines*, n.d.), in versions prior to 2.6.0 had a default configuration file that allowed remote connections and accepted unauthenticated access (Kadlec, 2017). Although administrators could correct this vulnerability by manually adding a line to the mongodb.conf file, this was not the default configuration and so many thousands of servers were left vulnerable (Kadlec, 2017). 60,000 MongoDB servers were left exposed online in early 2017 with around 57,000 servers attacked by various cyber criminals with databases wiped and/or ransomed (Cimpanu, 2020). Cimpeanu (2020) believes that most of the time these servers are exposed through administrators following incorrect MongoDB configuration tutorials, make regular configuration mistakes or use server images that come pre-packaged with misconfigured configurations of MongoDB out of the box.

Mitigations

Mousa, Karabatak and Mustafa (2020) recommend that systems are not expected to have any default accounts and that administrators should be highly trained and experienced.

Figure 2.8.4 shows an abbreviated example interactive prompt by a common shell script for Unix systems that is meant to improve security defaults for MySQL-like Databases. In the prompts we can see a user being asked to remove the anonymous users, disallowing root logins from the network as well as dropping the test database.

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!
```

Figure 2.8.4: Example of a prompt from the mysql_secure_installation that is a symlink to the binary mariadb-secure-installation shell script (MariaDB, n.d.)

2.8.3 Database Platform Vulnerabilities

Threat

Platform vulnerabilities are underlying vulnerabilities in operating systems (e.g. Windows, Unix, etc,.) and the additional services installed on a server hosting a database that may lead to exploitation and unauthorized access, data corruption or DoS conditions. (Oracle White Paper, 2017; Al-Sayid & Aldlaeen, 2013; Singh & Rai, 2014, Shulman 2006; Sarmah, 2019). For many companies the correction process to patch their server usually lasts between six to nine months (Sarmah, 2019).

Mitigations

Preventing these platform attacks, the literature suggests a combination of regular patching (software updates), incorporation of intrusion prevention systems (IPS) (Singh & Rai, 2014). Singh and Rai note though that vendor updates eliminate many vulnerabilities, organizations patch these systems in periodic update cycles, and in between these update windows the systems remain vulnerable. Sometimes updates are not feasible, even if needed for security, as there are compatibility problems (Sarmah, 2019; Singh & Rai, 2014). For this, Singh and Rai reiterate the need for IPS to monitor attacks on database traffic.

2.8.4 Privilege Elevation

Threat

Privilege elevation attacks occur when a threat actor takes advantage of database platform software vulnerabilities, such as built in functions, stored procedures, protocol implementations, SQL statements or other platform related vulnerabilities, to convert privileges from ordinary user to administrator (or other high privilege user) (Singh & Rai, 2014; Sarmah, 2019). This can result in all kinds of consequences such as transfer of funds, extraction of data, data destruction, data manipulation, deactivation of audit mechanisms, creation of bogus accounts etc. (Basharat, Azam & Wahab Muzaffar, 2012; Singh & Rai, 2014; Pevnev & Kapchynskyi, 2018; Mousa, Karabatak & Mustafa, 2020).

Mitigations

Query level access control privilege elevation can be prevented with intrusion prevention systems (IPS) inspecting database traffic to identify patterns which correspond to the behavior of vulnerabilities (Singh & Rai, 2014) as well as implementing through audit trails and enforcing principles of least privilege, to limit broad permissions (Pevnev & Kapchynskyi, 2018).

2.8.5 Privilege abuse

Threat

There are two subsets of privilege abuse, one legitimate and one excessive (Singh & Rai, 2014; Sarmah, 2019). Excessive privilege abuse is when users or applications that are granted database privileges abuse their privileges for malicious purposes outside of their job function or role scope (Singh & Rai, 2014; Pevnev & Kapchynskyi, 2018; Mousa, Karabatak & Mustafa, 2020). An example of this phenomenon would be a university employee whose job function only requires the ability to be able to change personal information, e.g. contact information, but takes advantage of excessive database privileges to change grades. This can happen because the database administrator has an ill-defined access privilege control mechanism, e.g. too many users in too broad permission groups (Singh & Rai, 2014; Pevnev & Kapchynskyi, 2018).

The other subset of privilege abuse is legitimate privilege abuse. This involves the user accessing data in a way that is normally within their job function but becomes unauthorized (Singh & Rai, 2014; Sarmah, 2019). An example of this would be a health care worker usually allowed to access their active patients, decides to access other patients, or manages to find a way to export large parts of the database to be able to work on their client machine.

Mitigations

Query level access control, a more granular access control that extends to specific rows and columns inside the database to only allow for certain SQL operations to be executed (SELECT, UPDATE, etc.). This can stop and help alert when the example university employee tries to change a student's grades (Singh & Rai, 2014). Good audit trails to record malicious behavior (Pevnev & Kapchynskyi, 2018).

2.8.6 Denial of Service

Threat

DoS/DDoS attacks aim to slow down the server, or servers serving the servers contents, and make it inaccessible to all users (Mousa, Karabatak & Mustafa, 2020; Sarmah, 2019). The difference between a denial-of-service attack (DoS) and a distributed denial of service attack (DDoS), is that a DoS attack originates from one machine, while a DDoS attack consists of multiple machines operating together, usually leveraged by a botnet (CISA, 2021). From a defending point of view, DDoS attacks are more difficult to deal with. When attacking with multiple machines, it allows for exponentially more traffic to be sent with flooding purposes. It also becomes harder to identify the source of the attack compared to if it were a singular machine sending the traffic (CISA, 2021). Though the technique itself does not expose contents of the database it can cause the victim reputation, time, and resources (Mousa, Karabatak & Mustafa, 2020). Shulman (2006) writes that common DoS techniques related to databases may be achieved by taking advantage of a database platform vulnerability to crash a server, network flooding, and server resource exhaustion (CPU, memory etc.).

Mitigations

Mousa, Karabatak and Mustafa (2020), Malik and Patel (2016) and Pevnev and Kapchynskyi, (2018) suggests some database centric mitigations with decreasing time of relations creations, TCP/IP stack reinforcements to allow and increase in the size of the TCP connection queue, employ Intrusion Detection Systems to detect and respond to attacks. Shulman (2006) also recommends dynamic profiling to detect unauthorized queries that may lead to a DoS such as vulnerabilities as well as connection controls to prevent server resource overload though limiting connection rates, query rates and other variables of each database user.

2.8.7 Weak Audit Trails

Threat

Threat actors maliciously accessing, editing, removing, injecting, or deleting data inside a database will leave traces and a robust audit trail will record these attempts or in worse case, successes (Singh & Rai, 2014). This can have the potential to seriously impact organizational risk, such as regulatory exposure (Al-Sayid & Aldlaeen, 2013; Sarmah, 2019).

Mitigations

Automated recording of all sensitive and/or unusual database transactions should be a part of all database security foundations (Singh & Rai, 2014; Pevnev & Kapchynskyi, 2018). This can help by providing deterrence if users are informed that their actions are being monitored (Sarmah, 2019), by meeting regulatory requirements for example HIPAA and GDPR by alerting and recording data breaches. If all other security measures have failed, robust audit trails can help detect intrusions/malicious use of the database and expedite recovery of the database (Singh & Rai, 2014).

2.8.8 Database Communication Protocol Vulnerabilities

Threat

Several security vulnerabilities have been identified over the years affecting database communication protocols leading to exploited databases, e.g., the SQL Slammer worm that exploded a flaw in Microsoft SQL server (Singh & Rai, 2014). These vulnerabilities have often not been detected by the native database audit control mechanisms, as they have not been designed to look at communication protocols (Singh & Rai, 2014; Sarmah, 2019).

Mitigations

The recommended mitigations to this threat are to employ protocol validation to parse (disassemble) database traffic and compare it to known good patterns, then it can act on these pattern matches to conduct alert or block actions on the traffic (Singh & Rai, 2014; Sarmah, 2019).

2.8.9 Buffer Overflow

Threat

A buffer overflow occurs when a buffer (area of memory allocated to a fixed or dynamic size) has more data copied to it then its allocated storage space can handle (Foster et al. 2015). Buffer overflows come in two flavors, heap overflow, where overflowed data corrupts the heap and may lead to exploits, and stack overflow where the buffers in the stack space has been overrun and the return address is overwritten allowing for arbitrary code to be executed (Foster et al. 2015). These buffer overflow attacks can lead to exploitation of the underlying platform the database is hosted on, the adjacent services or the database software itself (Al-Sayid & Aldlaeen, 2013) resulting in crash or server exploit (Shulman, 2006).

Mitigations

Writing secure code, auditing libraries and educating developers on best practices (Foster et al. 2015).

2.8.10 Attack on Backups

Threat

Mousa, Karabatak and Mustafa (2020), argue that database backup files are often left totally unprotected from attacks and that as a result of this frequent security vulnerabilities arise by database backup leaks. Backup storage media itself is often less secure compared with other database assets and with several high-profile data breaches involving theft or exposure of backup tapes and hard disks with backup copies of sensitive databases stored on them (Jain & Chawla, 2020).

Mitigations

Mousa, Karabatak and Mustafa (2020) and Sarmah (2019) recommend storing backup data in encrypted form to negate this threat. They continue to also recommend not overlooking

auditing servers and backup copies to look for indicators of attack and indications of compromise. Singh and Rai (2014) and Sarmah (2019) mention encryption of online production databases, but that they believe that it is too much of an overhead with performance and key management and that granular privilege controls are to be preferred.

2.8.11 Insider threats

Threat

According to Elmrabit et al. (2015) an insider threat is defined as "(a) Any malicious activities that cause damage to an organization's IT and network infrastructure, applications, or services - (b) On the part of an employee (current or former), contractor, subcontractor, supplier, or trusted business partner- (c) Who has or has had authorized access to the organization's IT assets - (d) And poses a significant negative impact on the information security elements (confidentiality, integrity, and availability) of the organization.". IBM (n.d.) expands this definition to also include infiltrators as inside threats as being "an outsider who somehow obtains credentials via a scheme such as phishing or by gaining access to the credential database itself". Insider threats are amongst the most common causes of database security breaches, attributing this to the fact that this is often the result of allowing too many users with privileged user access credentials (Chagarlamudi, Panda & Hu, 2009; IBM, n.d.)

Mitigations

Organizations can best defend themselves by employing comprehensive monitoring and analysis of not only behavioral indicators but of cyber and technical ones too (Greitzer et al., 2019).

2.8.12 Human error

Threat

According to IBM (n.d.) human error, leading to causing data breaches, can be accidents, use of weak credentials, sharing credentials or other unwise or uninformed behavior from users. A study conducted by Liginlal, Sim and Khans (2009) on 1046 data privacy breaches of U.S. companies between 2005 and 2008 showed that human error accounted for about 67% of all incidents and malicious acts only accounted for around 33% seen in Figure 2.8.5. IBM (2022) in their annual report of the cost of data breaches list human errors, being defined as "breaches caused unintentionally through negligent actions of employees or contractors", responsible for 21% of breaches.

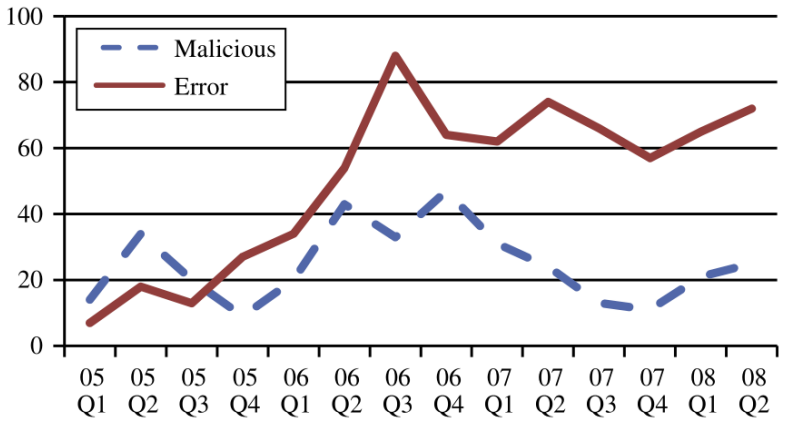

Figure 2.8.5: Trends in privacy breach incidents of all firms (Liginlal, Sim & Khansa, 2009)

Mitigations

Safa and Maple (2016) recommend tackling this threat by education, awareness, clear and understandable security policies, and organizational collaboration around information security.

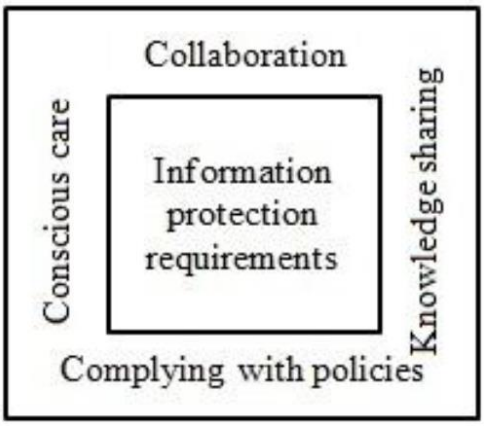

Figure 2.8.6: Information security requirements in organizations (Safa & Maple, 2016)

2.9 Security Practices

The purpose of this part of Section 2 is to give a very basic and fundamental understanding of some of the common best-practice solutions and frameworks in the field of database security.

2.9.1 Encryption

Classic Encryption

"Encryption provides a method of storing data in a form which is unintelligible without the 'key' used in the encryption" (Popek & Kline, 1979). In essence, traditional encryption can be represented by a function F, where D is the data to be encrypted, K is a variable key, and E is the resulting ciphered text (Popek & Kline, 1979). For the function to be of use, there must be

an inverse function which allows for the retrieval of the original data from the encrypted data, provided the initial key value is known.

$$E = F(D, K)$$

$$D = F'(E,K)$$

There has been substantial research in the field of encryption to develop algorithms, and they made it practically impossible to decrypt the data without having access to the key (Popek & Kline, 1979).

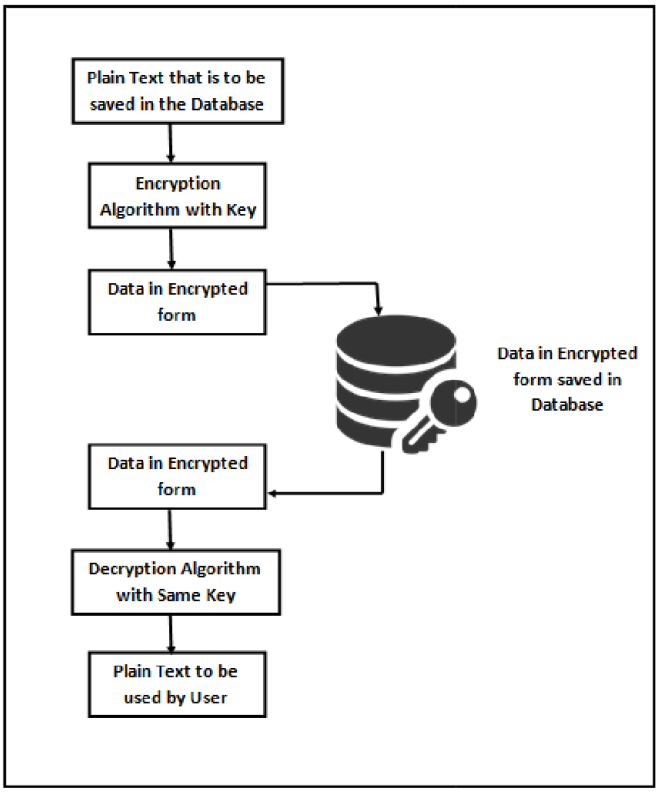

Figure 2.9.1: Database Encryption and Decryption Process (Singh & Kaur, 2015)

Encryption should not be mistaken for hashing as encryption is meant to be a two way process (Singh & Kaur, 2015) of converting plaintext into ciphertext with the help of a key (same key in the case of symmetric encryption algorithms or a private key of the private/public key pair in asymmetric encryption algorithms (Elmasri & Navathe, 2015) compared to hashing that is a one way process that takes plain text and converts it into a hashed value using a hash function, which cannot be changed back into plaintext (Singh & Kaur, 2015).

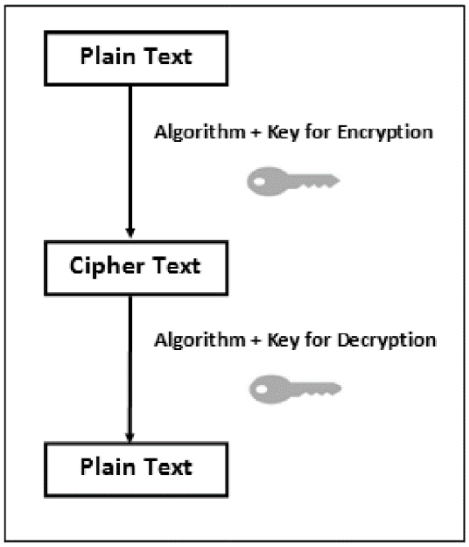

Figure 2.9.2: Working of Encryption Process (Singh & Kaur, 2015)

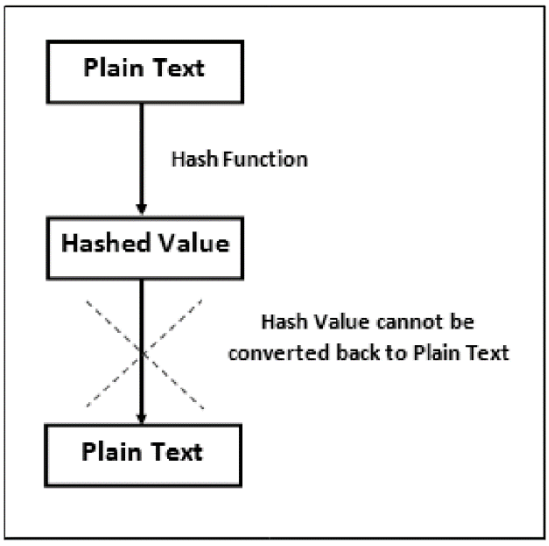

Figure 2.9.3: Working of Hashing Process (Singh & Kaur, 2015)

Encrypted databases are a popular approach to mitigate attacks on DBMS's (Grubbs, Ristenpart & Shmatikov, 2017). The threat of disk theft can be mitigated through Full Disk Encryption (FDE), but a disk theft on an Encrypted Database (EDB) without FDE still yields the persistent OS and DB state just not the volatile state. SQL attacks can enable arbitrary code injection with control over the memory space, thus yielding both persistent and volatile database states. Some virtual machine (VM) snapshots only store persistent states of the DB whilst others store both persistent and volatile states that include VMs memory and cpu registrars and this can lead to access to both persistent and volatile os and DB states. Full system compromise, or rooting the DBMS, also yields access to both persistent and volatile DB and OS states. Even though database encryption can be an effective mitigation it is not a full proof remedy to security threats (Grubbs, Ristenpart & Shmatikov, 2017).

Homomorphic Encryption

Homomorphic encryption (HE) schemes allow for data in databases to be searched, processed and analyzed without it needing to be decrypted (Gentry, 2009). This leads to mitigations against traditional database encryption pitfalls, being that data in ciphertext needs to be decrypted into plaintext before it can be processed and that decryption keys are often stored in volatile memory (Gentry, 2009). A database, encrypted with full homomorphic encryption, can now be stored on an untrusted server, such as a cloud server (IEEE, n.d; Chauhan, Sanger & Verma, 2015), but still allow for data processing without the data needing to ever be decrypted on the hosting server (Gentry, 2009). With more organizations moving their databases to the cloud, HE can be a tool to secure this data whilst still being usable (Røset, Warren & Chiang, 2017).

Successful homomorphic encryption schemes are based on highly complex mathematical problems and are deemed post-quantum, i.e., as of now safe against quantum computers cracking the encryption (IEEE, n.d.). A few areas where homomorphic encryption is seen as a solution are the sharing of genomic data, health care patient data and voting data to name a few (IEEE, n.d.).

There are some limitations and because HE implementations are still very computational heavy, systems must account for large computational overhead, multiuser database environments and not being standardized and user/beginner friendly (IEEE, n.d.).

2.9.2 Multi Factor Authentication (MFA)

"[A]uthentication is a process where a 'user identifies himself by sending x to the system; the system authenticates his identity by computing F(x) and checking that it equals to the stored value y" (Ometov et al., 2018). Ometov et al. (2018) argue that authentication is one of the most important safeguards when it comes to protecting devices or applications against illegitimate access.

MFA offers greatly improved security, as it forces people who are trying to access a system to provide evidence of their actual identity (Ometov et al., 2018). There are three factors available that you can use to "prove" your identity towards a service, these include:

1. Knowledge factor – something that only the person knows
2. Ownership factor – something that only the person has
3. Biometric factor – something that only the user "is"

(Ometov et al., 2018)

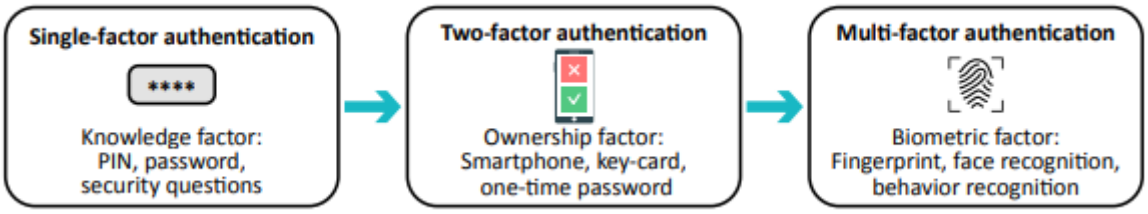

Figure 2.9.4: Multi-Factor authentication (Ometov et al., 2018)

The strongest practical technologies, tokens, certificates, biometrics, etc., and policies regarding authentication should be implemented to protect database authentication. Two-, or multi-factor authentication is preferred implemented together with directory integration for use of management and use (Shulman, 2006).

2.9.3 Zero Trust

The goal of zero trust is to "prevent unauthorized access to data and services coupled with making the access control enforcement as granular as possible" (Rose et al., 2020). Zero trust is an architecture that assumes that you never implicitly give trust to someone, instead you continually evaluate it (Rose et al., 2020). Every time you want to access a system, your identity needs to be authenticated and validated (Rose et al., 2020). Another important factor of zero trust is to always grant minimum privileges (Rose et al., 2020).

In figure 2.9.5 we can see how Microsoft 365 uses Attribute Based Access Control (ABAC) to enable zero trust by conditionalling access based on various signals from the user and the device connecting.

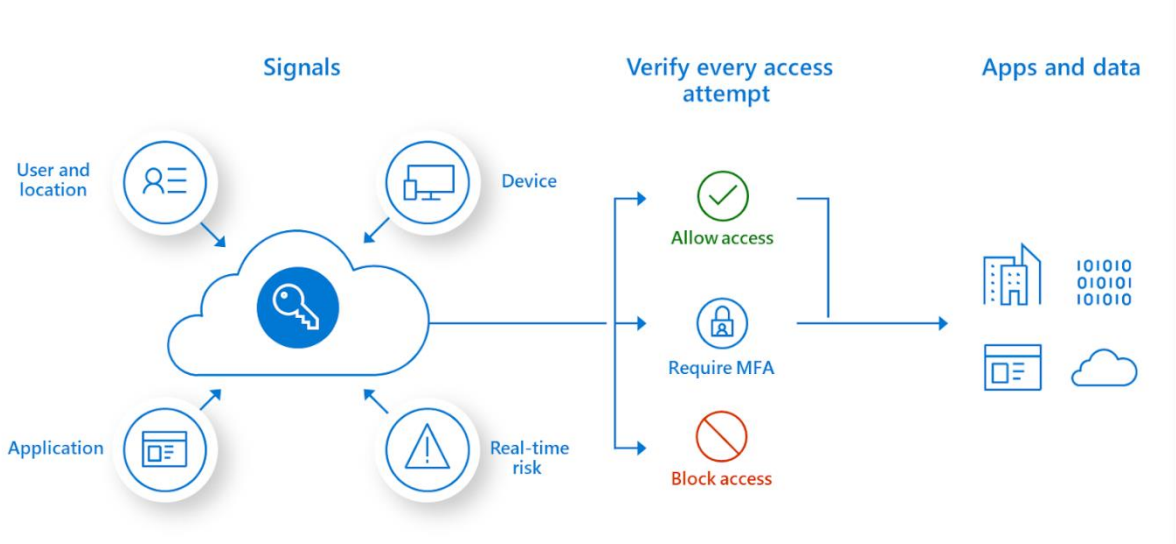

Figure 2.9.5: Zero trust example (Microsoft 365, 2019)

2.9.4 Hashing and Salting

Passwords are relatively easily compromised, which is problematic considering that they are the most used method of authentication in computer systems (Kelley et al., 2012). If a database is breached or leaked, the attacker suddenly has access to its data. If the database is storing hashed or encrypted data, the way to make it interpretable is to crack it (Kelley et al., 2012). Cracking is simply a word for the process of figuring out a password, and there are a few different techniques or technologies you can use (Educative, n.d.). Two common ways of cracking are brute force and dictionary attacks (Educative, n.d.). Brute forcing implies inputting any possible combination of letters and symbols to try and guess the password, while dictionary attacks instead use any possible combination of actual words (Educative, n.d.).

Hashing, using a cryptographic hash function (CHF) to generate a fixed-length hash based on input data with an arbitrary length, such as a password, can mitigate the risks of credentials leaking when a database is compromised (Menezes et al., 2018). CHFs have the property that it should be relatively easy to calculate the hash from the input but difficult or impossible to re-generate the original input if only the hash is known. Another property of CHFs should be difficult to create an initial input that would match a desired output (Menezes et al., 2018; (Owasp, 2018). There are a multitude of hashes to use in databases today, some older may have collision vulnerabilities (different sets of input data generate the same identical hash) (Black, Cochran & Highland, 2006; Stevens et al., 2017) or are too fast to calculate leading to easier cracking, whilst other provide such a cost to attackers that it is not feasible to mass crack credentials in stolen databases, see Table 2.9.1.

As people often reuse old passwords or use the same passwords as other people, the hash of these same passwords will be the same, so for a database containing several hashes that are identical an attacker only needs to spend energy to crack one hash and use that result to crack the others, a mitigation to this is salting (Arias, 2018). Salting hashes involves adding uniqueness to every single hash, so that even if two users use the exact same password the hashes will be dissimilar, adding cost of an attacker having to treat every hash as unique even if the passwords might be the same (Arias, 2018).

Hashing and salting do not however protect databases from online attacks, such as trying different credentials in a web portal, the mitigation for this is lockout times, anomaly detection and MFA (Arias, 2018).

As seen in Table 2.9.1 there are large differences between hashing algorithms, such as the older MD5, SHA1 and SHA2-256 with a modern GPU being able to perform billions of guesses each second and a modern approach such as bcrypt with many iterations only allowing for a modern GPU to guess hundreds, or maybe a thousand guesses a second. The magnitude of difference between MD5 and bcrypt(shal512($pass)) with 4094 iterations is an astonishing 8839160000% change, an order of billions longer to crack a password with a better modern crypto.

Hashcat v6.2.6; GPU = Nvidia GeForce RTX 4080; CPU = AMD Ryzen 9 5900X 12-Core

Hash Type	GPU	CPU	Diff %
MD5	91573.7 MH/s	2282.7 MH/s	97.5%
MD5($salt.$pass)	91501.2 MH/s	2376.9 MH/s	97.4%
SHA1	29733.1 MH/s	1336.7 MH/s	95.5%
SHA1($salt.$pass)	29868.5 MH/s	1332.4 MH/s	95.5%
phpass, WordPress (MD5), Joomla (MD5)	26760 kH/s	820.4 kH/s	96.9%
SHA2-256	12870 MH/s	552.9 MH/s	95.7%
sha256($pass.$salt)	12705.8 MH/s	549.3 MH/s	95.6%
SHA3-256	2833.1 MH/s	153.4 MH/s	94.5%
SHA3-512	2814.3 MH/s	154.2 MH/s	94.5%
(bcrypt(sha1($pass)) / bcryptsha1) [Iterations: 32]	121.0 kH/s	16172 H/s	86.6%
(bcrypt(shal512($pass)) / bcryptsha512) [Iterations: 4096]	1036 H/s	138 H/s	86.6
(PBKDF2-HMAC-SHA1) [Iterations: 999]	10999.5 kH/s	521.3 kH/s	95.2%
(PBKDF2-HMAC-SHA256) [Iterations: 999]	4559 kH/s	194.0 kH/s	95.7%
(MSSQL (2012, 2014))	3710.3 MH/s	175.2 MH/s	95.2%
(PostgreSQL SCRAM-SHA-256) [Iterations: 4095]	1256.8 kH/s	47996 H/s	96.1%

Table 2.9.1: A comparison benchmarking popular hashes using Hashcat on a modern GPU and CPU.

There are however more specialized tools such as Field Programmable Gate Arrays (FPGAs) that can be programmed for a special application, such as being optimized for a single hashing algorithm (Nardi, 2020; van Beek & Gevers, 2020). Table 2.9.2 shows a comparison of how efficient FPGAs can be compared to modern relative GPU/CPU architectures, note also the wattage use, as a comparable RTX 2080 rig of 75-80 cards would consume around 25 kilowatts of power (van Beek & Gevers, 2020).

Hash / Work Factor	18 x ZTEX 1.15y (FPGA) 585 Watt	1 x RTX-2080Ti (GPU) ~330 Watt	1 x AMD EPYC 7401P (CPU) ~270 Watt
bcrypt / 05	2,100,000 H/s	28,000 H/s	25,200 H/s

Table 2.9.2: Comparison of cracking speed of FPGA vs GPU vs CPU on bcrypt (van Beek and Gevers, 2020).

2.9.5 Centralized & Federated Identity Management Systems

There are three main models of identity management systems (IdMs), the isolated model, the centralized model and the federated model (Carretero et al., 2018). With isolated IdM the service provider (SP) and identity provider (IdP) are combined on a single server as shown in Figure 2.9.6. This is a very simple and common approach (e.g., a webshop hosting its own user database for authentication) but with the increased growth of online services user will have to manage very many different credentials (Carretero et al., 2018). If users use different credentials for each service (as they should) there may be a lack of usability, on the other hand if they use the same credentials for many services there may be a security risk.

The second Figure, (Figure 2.9.7) shows a centralized IdM which enables users to only know or have one credential to access many services. A well-known implementation of this model is the use of it in Windows Active Directory environments through Keberos (Carretero et al., 2018). Single Sign-on (SSO), an implementation method of the centralized IdM, is often used today to both enhance security, though SPs not having to know users' credentials and ease of use for the user only having to know one credential. A possible downside of the centralized IdMs is the potential for a single point of failure (Carretero et al., 2018).

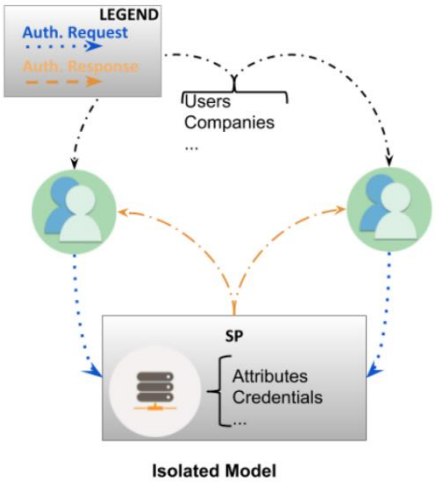

Figure 2.9.6: The isolated IdM model (Carretero et al., 2018)

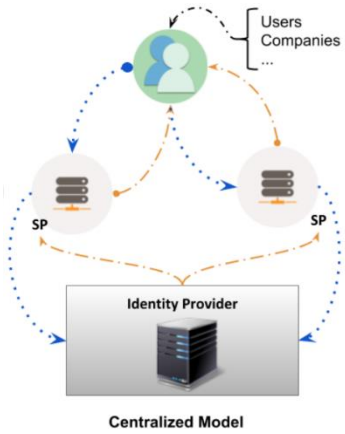

Figure 2.9.7: The centralized IdM model (Carretero et al., 2018)

The federated IdM model is an evolution of the centralized model as it allows for implementation in heterogeneous topologies by letting the parties involved to agree to which entities should be part of the system, how they are going to be referred to and how the configuration should look like (Carretero et al., 2018). This allows for authentication across independent domains and SPs though different IdPs all while getting rid of the single point of failure, see Figure 2.9.8. The federated approach allowed for a SP to confirm the identity of the user (Authentication), allowing or denying access to a resource based on their attributes (Authorization) as well preserving anonymity of the user by usage of anonymous identifiers establishing an association with the local identity (Carretero et al., 2018).

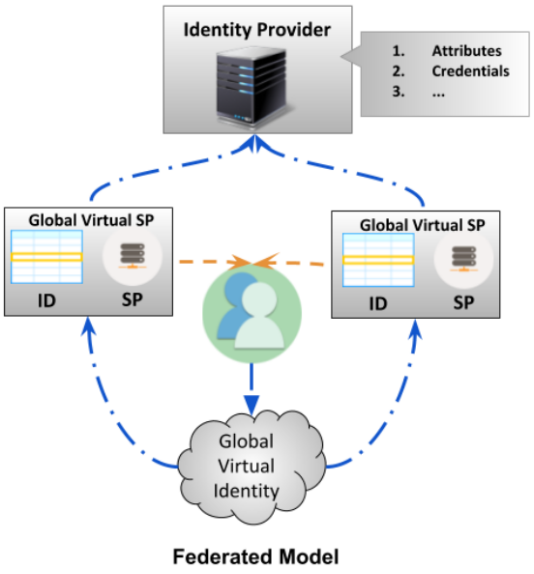

Figure 2.9.8: The federated IdM model (Carretero et al., 2018)

3 Methodology

3.1 Qualitative study

The purpose of this study is to understand the past in order to secure the future of database security. To be able to answer our research question, insight is needed from security experts with relevant and long experiences in the field. A qualitative study in the form of interviews provides relevant and thorough data that can allow for an in-depth analysis into a topic (Jacobsen, 2002). This together with our research question led us to choose the qualitative approach over the quantitative one. Rubin and Rubin (2005) provide a good analogy to qualitative interviews and that is that they function like night goggles, ''permitting us to see that which is not ordinarily on view and examine that which is looked at but seldom seen'' (Rubin & Rubin, 2005, p. vii). We expected experts within the industry to have wildly different experiences and opinions based on factors such as where they have worked, what type of incidents they had been exposed to etc. It would be difficult to make any form of standardized survey for a quantitative approach and get useful data from it. Our choice of method lets us roam outside our initially scoped questions to get relevant and correct data in the unique context of the interviewee's individual experiences in the hope of finding common denominators across the board in the final analysis.

3.2 Selection of Interview Participants

To be able to successfully answer our research questions, we needed our interview participants to have at least 15 years of experience in the field of cyber security, but preferably more such as > 20 years. This, as well as the other two criteria are listed below.

The criteria for informants were:

- **At least > 15 years, or preferably > 20 years of experience in cyber security.**
- **Most of that time working or otherwise active in IT-security, IS-security or cybersecurity in Sweden.**
- **Exposure throughout their careers to Swedish databases.**

To get in contact with such candidates, we used three methods. The first one was to utilize our own social networks. We have an acquaintance who fit the profile that we needed for an interview person, and they were the first person we reached out to. We soon found out that many experts in the field, especially those based in Stockholm, know each other fairly well. Our first confirmed interviewee connected us with experts in the industry, and we asked those we thought fit our profile to be interviewed about their experiences. Using snowball sampling (Noy, 2008) we found more potential candidates through people's networks. The second method was for us to ask the organization we work at to connect us with their security personnel. Our third and last method was to search on various platforms such as LinkedIn and university contact pages for people with the desired experience and contacting them through email. We deliberately tried to obtain interview candidates from different backgrounds and organizations, as to not have, for example, four interviewees from the same organization, as

we don't believe that it would justly reflect the landscape of national database security in Sweden.

During our planning we became aware that an interview of this nature opens for potentially sensitive information to be discussed, even if our scope doesn't necessarily entail going into any specific organizational strategies. Based on this we offered all our interview participants the option to partake in the interview anonymously.

All candidates, after they had agreed to be interviewed for our study, were provided with an email with the interview introductory information as well as the questions. Meetings were booked in advance of typically one week prior to the interview and the time allotted was one hour during office hours. According to Oates (2022) an ideal interview time is 30 minutes to one hour. We tried to set a goal for ourselves to have the interviews last about 45 minutes, leaving around 15 minutes for small talk and introduction. Table 3.2.1 lists the interviewee details.

Appendix	**Name**	**Organization** *Past experience*	**Role**	**Experience in security**	**Interview length, method & date**
C	Lars Otterskog	Polismyndigheten *Government security agency: SÄPO as* well as a consultant in the private sector	Security Specialist	18 years	47:30, Teams meeting, April 19, 2023
D	Anders Hjortberg	Tetra Pak *Software Engineer* as well as *Swedish Armed Forces IT/IS manager*	System Infrastructure Specialist	25 years	37:16, In person, April 20, 2023
E	Jesper Blomström	Cparta Cyber Defence *Government security agencies: FRA and SÄPO as* well as a consultant in the private sector	Red Team Manager	15-20 years	33:15, Teams meeting, April 21, 2023
F	Christoffer Jerkeby	Jerkeby Security AB *Security researcher and consultant in the private sector*	CEO	20 years	1:14:03, Teams meeting, April 24, 2023

Table 3.2.1: Interview participants

3.3 Interview process

Our interview questions were formulated to be able to effectively address our research questions. Due to the fact that our empirical data would stem from people's accumulated experiences, we required genuine responses with strong validity. Our aim was to develop a set of questions that were as open-ended as possible, because these can help in maintaining high internal validity during interviews (Jacobsen, 2002). We believe that our interviews were conducted as semi-structured interviews with a list of questions as well as asking additional questions if the interviewee brought up issues we believed were of interest (Oates 2022). This would hopefully allow the interviewees to speak with more detail on issues we would raise and as well introduce issues of their own that they would think were relevant to our theme (Oates 2022). Following qualitative interview best practices suggested from Myers and Newman (2007) we tried to structure our interviews following their model (Figure 3.3.1).

These are:

- situating the researcher as actor (gain background information to help readers assess validity of the findings),
- minimize social dissonance (manage first impressions, dressing appropriately and using appropriate language/jargon),
- represent various voices (interviewing a variety of people from different backgrounds, "triangulation of subjects" (Rubin & Rubin, 2005, p. 67)),
- everyone is an interpreter (recognize that subjects are creative interpreters of their worlds, as we are of theirs),
- use mirroring in Q&A (using words and phrases used by subjects to construct follow up questions and comments, letting them explain their world in their own language rather than imposing ours, using open questions),
- flexibility (these being semi-structured interviews, be prepared to explore interesting avenues of research),
- confidentiality of disclosures (keeping raw data and transcripts confidential and secure) (Myers & Newman, 2007).

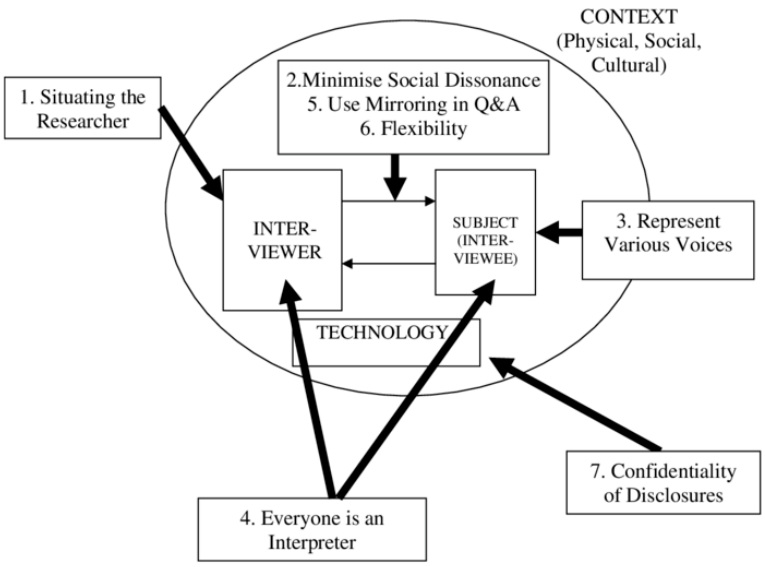

Figure 3.3.1: Guidelines for the qualitative research interview (Myers & Newman, 2007).

All but one of our interviews were conducted through Microsoft Teams meetings with cameras on, and the remaining took place at the interviewees organizational work office. An important factor for our model was the ability to be able to see the interviewee. There are physical signals during an interview such as the interviewee looking noticeably uncomfortable or threatened (Jacobsen, 2002). By using Teams meetings as a tool over for example a standard telephone call, we minimize the risk of thereby affecting the results in itself, however it also lets us steer the interview a certain way if we notice some of these signals.

Interviews were started by thanking the candidate for their participation as well as informing them of their right to withdraw from the interview at any time, changing or rephrasing an answer at a later stage, the opportunity of removal of their interview as a whole or in parts before publication, as well as full or partial anonymity (name, title, organization or place of interview). The interviewee was also informed that the audio recording would be destroyed after transcription was completed and not to be shared outside of this study. After agreeing to be interviewed and informing the interviewee of the introduction information (see 3.5 Ethics) the recording of the interview would start after their consent. The interviews were conducted in Swedish.

The interview recording ended after the interviewee provided their answer to the final question. After the recording was stopped the interviewee was thanked again for their participation and offered to read the thesis when completed.

3.4 Transcription

Following advice outlined by Oates (2022) we transcribed our interviews into text enabling us to perform analysis on textual data. When transcribing the interviews, we again followed recommendations outlined by Oates (2022) and chose to format the text from each interview in a similar manner in tables with rows separating questions from answers as well as leaving columns with spaces to allow for coding and location markers. Author questions, follow up questions and comments were written in bold. The predetermined interview questions, numbered 1 to 16 in the interview guide, that were asked by the authors were included in the transcript. Also, all follow up questions were transcribed. The pre-interview information (see: Appendix Part A) was omitted from the transcript as this was not recorded. We only started recording audio after the participants were informed about their rights (see: 3.6 Ethics) and information about the study and gave us their informed consent to participate and be recorded.

The authors chose to include all answers from the interviewee in the transcription except for temporary stuttering, sighs, and filling words such as "hmm", "jadu" as well as false starts and unintelligible sounds. According to Oates (2022) this is a common way of transcribing interviews for researchers. We also chose to remove similar filling words and sounds from the interviewers but otherwise tried to transcribe the whole interview as accurately as possible even if some sentences were grammatically hard to read. We admit there may be some informational loss as to losing tone, pitch, intonation as well as non-verbal clues when transcribing meetings, both video and physical ones, to a written transcript. Although Oates (2022) gives examples on conventions about how to show such aspects in a transcription we decided it was not critical to our study and opted out using them.

Each interview audio was recorded digitally on the interviewer's computer except the one interview that took place at the interviewee's office, where audio was recorded on the

interviewers' phones. The process of transcribing was that of listening to the recorded audio and as faithfully as possible manually writing down the conversation in the appendix of this thesis. As the interviews varied in length, from 33:15 to 01:14:03, as well as varying degrees of microphone quality from the interviewee, the transcription time took from one to three days per interview.

3.5 Data analysis and coding

Once all interviews were transcribed and the texts placed inside of tables, we followed Oates (2022) recommendations of first reading through all of our data to obtain general impressions. Both authors conducted the read through of the transcripts together, discussing the contents and identifying key themes throughout the interviews. Initially we used just three themes, again following Oates (2022), those were:

- Segments that bore no relation to our overall research purpose.
- Segments that provided a general descriptive information that we thought could be needed to describe the research context to our reads (for example, experience in cyber security, working in state security services versus the private sector)
- Segments that appeared to be relevant to our research questions.

After our first read through we then, again based on Oates (2022) recommendations, looked at the parts that seemed to align with the third segment, text that appeared to be relevant to our research questions, and tried to find themes corresponding to themes in our background theory (see: 2 Theoretical Background). This theming was also conducted on categories that we believed we observed in the data, known as an inductive approach (Oates, 2022). This was done with color coding according to theme to make it easier to find relevant information (see: table 3.5.1: Color Codes). We also chose to code segments of interest that had a connection to the CIA-triad (see table 3.5.2: Code).

In the last phase of the analysis we began to compare similarities and differences from the answers from the interviews.

Subcategories	Color Code	Aspect	Code
Experience	Red		
Historical	Yellow	Historical events or context	HIS
Policy	Green	ISO 27000	ISO
		GDPR	GD
Threats & Challenges	Orange	SQL-Injections	SQL
		Misconfigurations	MC
		Platform Vulnerabilites	PV
		Priviliage Elevation	PE

		Privilage Abuse	PA
		Denial of Service	DoS
		Weak Audit Mesures	WA
		DB communication protocol vulns	CPV
		Buffer Overflow	BO
		Attack on Backups	AoB
		Insider Threats	InT
		Human Error	HE
Mitigations	Blue	Encryption	ENC
		MFA	MFA
		Zero Trust	ZT
		Hashing & Salts	HS
		Federation	FED
Future Trends	Light Green	Future trends of threats and mitigations	FT
Culture	Purple		CUL
Education & Awareness	Cyan	Education	EDU
		Awareness	AWA

Table 3.5.1: Color Codes

3.6 Approach to Literature Selection

For our thesis we sought to find literature to help define and provide context to our problem statement and research questions. Quite soon we noticed that finding multiple sources of high-quality literature specifically for database security was hard to find, high quality literature defined by us as being research papers, conference submissions and other peer reviewed research. To address these issues, we also accepted other sources of information such as technical standards and best practices published by national and international standards organizations. We found that this inclusion did not either totally fill the void of information. As a lot of cyber security research and practices are funded and produced by non-academic institutions, we also included sources from companies, governments, agencies and groups not connected to academia (Gernhardt & Gros, 2022). We also found that some information covering events such as database breaches were only covered by media outlets and journalists, so this too was included. We adopted a ranking system from Guptill (2016) in which we would look for information, going from the top down until we found a good source to use (see: Table 3.6.1: Literature ranking). Because Guptill did not focus on information systems, we also

added an abstract tier for ourselves above tier one, we can call it tier 0, in which we sought to find good sources first and foremost from the basket of 8 publications.

Tier	Type	Content	Uses	How to find them
1	Peer-reviewed academic publications	Rigorous research and analysis	Provide strong evidence for claims and references to other high-quality sources	Google Scholar, library catalogs, and academic article databases
2	Reports, articles, and books from credible non-academic sources	Well researched and even-handed descriptions of an event or state of the world	Initial research on events or trends not yet analyzed in the academic literature; may reference important Tier 1 sources	Websites of relevant agencies, Google searches using (site: *.gov or site: *.org), academic article databases
3	Short pieces from newspapers or credible websites	Simple reporting of events, research findings, or policy changes	Often point to useful Tier 2 or Tier 1 sources, may provide a factoid or two not found anywhere else	Strategic Google searches or article databases including newspapers and magazines
4	Agenda-driven or uncertain pieces	Mostly opinion, varying in thoughtfulness and credibility	May represent a particular position within a debate; more often provide keywords and clues about higher quality sources	Non-specific Google searches

Table 3.6.1: Literature ranking (Guptill, 2016)

When looking for good credible tier 1 sources we used academic search engines listed below. We searched for keywords such as "Database Security", "Database Vulnerabilities", "Database Breach", "Database Security Policies", "Database Security and Integrity", "Secure Database Design", "History of Database Security", "Svenska Databasincidenter", in academic search engines such as Elseviers ScienceDirect, IEEE Xplore Digital Library, LUBsearch, Google Scholar

3.7 Research Quality

3.7.1 Reliability

Selltiz et al (1976 cited in Brink, 1993) argues that reliability is concerned with the consistency, stability and repeatability of accounts given by an informant as well as the researchers ability to collect and record data in an accurate way.

Robson and McCartan (2015) suggest some practices to follow when conducting qualitative research and that is "being thorough, careful and honest in carrying out the research, but also being able to show others that you have been" (Robson & McCartan, 2015, p.173). Robson states, comparing qualitative research to quantitative and observational standardized research instruments, that "[a]t a technical level, the general non-standardization of many methods of generating qualitative data precludes formal reliability testing." (Robson & McCartan, 2015. p.173) and continues to add that there are pitfalls to all types of data collection and transcription such as equipment and transcription errors as well as interview environment distractions and interruptions. Jacobsen (2017) and Myers and Newman (2007) provide examples of

threats and their mitigations towards reliability, these are: interviewer effect (the interviewee is affected by the interviewers' characteristics; clothing, age, body language, jargon, attitude etc.), though impossible to fully mitigate the interviewer effect we tried to follow Myers and Newmans (2007) recommendations and acknowledge this properly effect every interview to some extent.

A potential challenge is artificiality of the interview, i.e., is the interview conducted in a familiar and "safe" setting for the interviewee where she feels at ease? Or is the interviewee interviewed in a strange setting, feeling pressured to give answers to complicated questions under time pressure? We tried to solve this by interviewing people following their schedule, allowing for extra time, and from their own offices and homes on Teams and the one physical interview was conducted at the interviewees' companies' offices.

Another challenge is planned or surprise interviews (depending on if the interview is planned or spontaneous the data will properly differ, the first probably eliciting more planned and thought-out answers and the later probably better spontaneous feelings and opinions). All our interviews were planned with interviewees receiving copies of questions days before the interview so that they could feel relaxed and well prepared to gather their thoughts.

Furthermore, another mitigation is to consider carelessness in recording and analyzing data. However good the information from informants is we can never get better data than what we manage to record and analyze. To mitigate this we were careful to record every interview with good quality software tested beforehand (when conducting Teams interviews) as well as recording on separate devices during the physical interview and we reviewed each other's transcriptions. Together we coded and categorized the transcriptions to facilitate finding relevant data. We could have improved on our process by allowing an external researcher or subject expert conducting an independent coding and categorization but unfortunately this did not materialize.

3.7.2 Validity

Robson and McCartan (2015) write that validity in qualitative research is something to do with it being accurate, or correct, or true. Robson and McCartan (2015) and (Jacobsen, 2017) offer several suggestions on how to mitigate "threats" to validity. Robson and McCartan (2015) provide three main types of threats to validity being *description, interpretation, and theory*. Providing a correct and valid description of what we have heard from our interviews is paramount and Robson and McCartan (2015) suggest that audio- or video recordings should be carried out, as we have done during all of our interviews. The threat to providing a valid interpretation, Robson and McCartan (2015) argue, is through imposing a framework or meaning on what is happening in the study instead of letting this arrive from what we learn during our involvement in the study. Mason (1996 cited in Robson & McCartan, 2015) stresses that we as authors should not take our interpretation of our collected data as self-evident but instead continually justify our steps through which our interpretations were made. We have, during our study, kept an open mind to the direction of where it might take us and have had continuous discussions about the findings. The last threat Robson and McCartan (2015) describes is the threat of not having a valid theory, and he suggests that we consider alternative explanations of what we are studying by seeking data that is not always consonant with our set out ideas. We have, during our thesis development, sought to find theories that counter, or in some way contradict our findings. This together with finding informants from

different backgrounds and sectors, we hope to achieve triangulation to counter any validity threat (Robson & McCartan, 2015).

Jacobsen (2017) writes about two main categories of validity in qualitative research. The first is internal validity and concerns the validity of gathered data. Did we find the right sources and did these sources provide us with reliable information? We created some criterias potential informants had to fulfill, e.g., at least > 15 years, or preferably > 20 years of experience in cyber security. In reaching out to our first contact who met these criteria and is a senior cyber security researcher who has lectured both domestically and internationally, we were confident of finding other candidates through him using snowball sampling (Noy, 2008). We were also relatively certain that the candidates found could provide reliable information as they also fulfilled the other criteria we had except the years of experience, notably exposure throughout their careers to Swedish databases. We are also aware that we ourselves became more familiar with the topics in this research the more literature we read and, maybe to an even greater extent, for every interview held. Jacobsen (2017) tells us that some researchers regard that data which is collected later in the process is of a higher quality, because we ourselves gain knowledge and can ask better questions, but Jacobsen tells us that the downside of this is that we as researchers can be blinded by this and that we only search for information that supports the assumptions we have already made. To counter this potential bias, we firstly made a concerted effort to not change interview questions, or more precisely comments or follow up questions too much, as to not invalidate the result of different interviews from each other as well as to keep comments and other small talk to an absolute minimum during the interviews to keep them as uniform as possible.

The second category of validity Jacobsen (2017) writes about is external validity which is about the degree of which the results from the study can be generalized to others than those interviewed. Jacobsen continues to explain the reason qualitative research often only examines few units is because they often offer interesting perspectives or have sought after information. The downside with this approach, Jacobsen argues, is that it can be hard to claim that this small selection is representative of a larger population of units. Jacobsen continues to provide two conditions for generalizing the data provided by informants, the first being the number of informants (units) and the second how the informants (units) have been selected. We started by reaching out to several candidates, once we had conducted four interviews and were satisfied with their level of experience and we believe we achieved enough saturation. The four informants, seen in Figure 3.7.1 (A=authors, R*=informants in order of contact) did not all know each other, nor did all have the same background and general experience such as if they had come from the same school, organization, or company, therefore we felt we had achieved a good spread of informants.

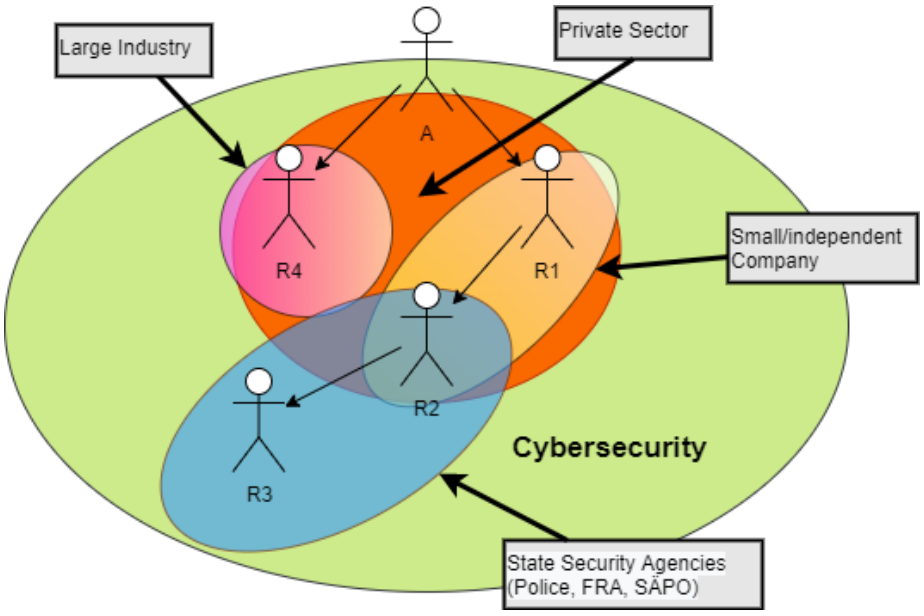

Figure 3.7.1: Spread of informants, their main background, and social connections.

3.8 Ethics

Oates (2022) states that it is imperative to treat everyone involved in our research "fairly, with honesty and with an eye to duty of care". Oates (2022) summarizes this as we, as the authors of any research, should be an ethical researcher. Oates (2022) lays out five key rights of participants involved in any study and we as authors have strived to respect these rights of our participants.

The first right is the right for the participants not to participate in our study (Oates 2022). When contacting potential candidates to participate in our interviews we attempted to use polite and professional language. We did not try to pressure or persuade any participants if they were not interested in participating, and we did not try to hound potential participants if we did not get a response from them. If we reached out by email with no response, we only tried to reach out to their LinkedIn once with no further attempts of contact. We were fortunate that most participants we received a response from were interested in participating, the few who did not want to participate were thanked for their time and were not persuaded further.

The next right for the participants that Oates (2022) lists is the right for a participant to withdraw from the study, that for someone initially agreeing to participate can change their minds at any time. Oates (2022) continues that this right also extends to participants having the right to decline to answer certain questions. Through informing the participants in the pre-interview information that "The candidate can also choose to cancel the interview at any time or withdraw any part/parts or all of the interview." We hope to inform the participants that they could cancel or withdraw from the study/interview at any time as well as that we would delete any answer, they were uncomfortable with.

Oates (2022) states that the third right of the participant is the right to give informed consent. Oates (2022) continues that this informed consent can only be given if the participants have been made fully aware of the nature of the research, their involvement and the intended use of the research as well as the following; the purpose of the research, who is undertaking the

research (name, contact details) and which organization is overseeing and authorizing the research (in our case, the Department of Informatics at Lund University School of Economics and Management), what will be asked of participants (in our case: hour long interview), if they will receive any compensation or incentive (for example a copy of the published thesis and its findings, although this was offered by the offers only after the interviews were finished), how their data will be stored (we informed all participants before starting the interview that audio would be recorded during the interview and that we would destroy this data after transcribing it).

The fourth right that Oates (2022) lists is the right of anonymity for the participants. As part of our pre-interview information, we gave the participants the opportunity to have parts or all of their attributes anonymized. This included their name, title, organization and location of the interview. During the interviews we sometimes moved on from a question that we felt the interviewee had a hard time answering without breaking the confidentiality of a contract or law, as not to push them into exposing themselves to risking divulging confidential or secret information. We also interjected during interviews and asked if they wanted us to not transcribe something that could have the potential of exposing sensitive information.

The last right Oates (2022) lists is the right to confidentiality for participants. Oates (2022) goes on to explain that confidentiality includes both data, as an example in the form of recordings in our case, in publicly accessible spaces and that it is our responsibility to store any such collected data safely. We informed the participants that all recordings would be deleted after they were transcribed. Oates (2022) also explained that this confidentiality extends to something they might say in confidence to us or off the record. Even if we were to have interesting discussions in the small talk before and after the interviews, we did not use any of it in our research.

We provided information about the rights above (see: Appendix Part A: Interview Guide) in written form in both English and Swedish in emails to participants as well as read it out loud before beginning the interview. The participants had to give their verbal consent after the pre-interview information was read to them before starting the interview and audio recording.

4 Results

This chapter will entail the findings and results of our empirical qualitative study. Our study consists of four interviews with practicing experts in the Swedish cyber security field. Interview transcripts can be found in their entirety in the appendix section. Part C and onward are interview transcripts and these will be referenced in the following way "(Part C, answer 4)", representing the 4th answer of that interview.

4.1 Outdated threats & the lessons learned

Two of our informants answered that SQL injections were some of the most common threats against database security 20 years ago (Part E, answer 5 & Part F, answer 5). One of them explained that the problem area behind SQL-injections was that frameworks at the time allowed for string concatenation between user data and data from a database before a query (Part F, answer 8). In answer 6, he explains that when parameterization of SQL-queries was introduced, it led to them no longer being of relevance. Equally, a third participant also explained how the community learned a lesson regarding SQL-injections, and he also attributed the development of new frameworks and the fact that they forced the developers to parametrize the queries to the extent that SQL-injections no longer pose a relevant threat (Part E, answer 7).

Three informants mentioned storing credentials such as usernames and passwords in plaintext inside of databases as a major risk in the early 2000s (Part D, answer 5, Part E, answer 7 & Part F, answer 6). There was not much further talk about this subject in particular, except for one participant who briefly added "That was stupid." after mentioning plaintext storage (Part F, answer 6).

4.2 Evolution of threats

One informant explained that there was a noteworthy underground hacking movement in Sweden from 2007 to 2010 and stated that there were many databases being leaked in a short period of time back then (Part F, answer 7). He proceeds to explain that people stopped hacking without any financial incentives around 2010, as well as a split between the hackers from an academic background and the more underground hackers, as the academic hackers were offered comfortable jobs like CISOs, and that that time period marked the end of "Swedish hacking" as a phenomenon.

The same informant proceeds to talk about how the criminal hacking world separated from one actor doing all of the work to having "initial access brokers", which are organizations that hack into services and sell databases or accesses in the form of tokens and password for money (Part F, answer 10). This phenomenon has, according to him, recently been legalized in Russia and has become legitimized to the point where you can even receive help from the state. When asked about how the threat picture has evolved, another informant also brought up the fact that state or state-sponsored groups are responsible for a lot of the attacks (Part E, answer 6).

Another important event in the evolution of the threat landscape was the commercialization of modern graphic cards, which were previously only available on a state level. (Part C, answer 8). Likewise, a second participant spoke about graphics cards when asked about how the threat landscape had changed (Part F, answer 7). The same participant explained that the reason the new era of graphics cards was so good at cracking passwords that used standardized encryption was the fact that the graphics cards often came with these algorithms already implemented.

"Suddenly, any gamer could use their graphics card and get a performance better than that of specially designed chip sets" (Part C, answer 8).

This sudden change swiftly led to a change in hashing algorithms, what had previously been deemed safe wasn't anymore. (Part C, answer 8).

4.3 Prominent incidents

Two of our informants talked about incidents directly affecting Swedish governments. One of the participants brought up the "carelessness" regarding the leaks from the Swedish Transport Agency and questioned if any lessons had been learned from it (Part D, answer 7). He spoke about how a lot of information around it had been quickly silenced, and how a minister was kicked out but later came back anyways. The other participant also brought up the same incident and explained that foreign staff got access to something that they shouldn't have, and that said staff wasn't properly security vetted (Part C, answer 7). In reference to this answer, he explained that one can have many technical protective measures in place, but if something goes wrong with access handling it instantly defeats the purpose of those technical measures.

"Security doesn't only have to be of technical nature, but it can also be processes and routines that you need to account for when it comes to accessing information." (Part C, answer 7)

The Swedish Tax agency was also victim to an incident where they got hacked and the perpetrator got access to their databases. Personally identifiable information was leaked on people with protected identities (Part C, answer 7). The Public Health Agency of Sweden also suffered an incident regarding a vaccination register (Part E, answer 7).

There were also some answers regarding non-state or state-organization incidents. "Gunneboläckan" is a leak of a particularly sensitive nature (Part F, answer 7). Gunneboläckan is an incident where the organization who hacked the company "Gunnebo" leaked all the accessed information online. Gunnebo is a company that manufactures and services bank vaults, secure rooms for prisons and other security products (Part F, answer 7).

"This is, of course, very sensitive. Sketches and the database dump leak onto the internet in connection with Gunnebo refusing to pay. This is a rather complex database dump that contains not only dangerous business information but also specific details." (Part F, answer 7)

The informant continues by explaining that this leak was an eye-opener. Since Gunnebo is a security company, he thinks the discussion shifted from them being at fault to the idea that it

could happen to anyone. And if it can happen to anyone, one must be certain about how to handle databases (Part F, answer 7)

4.4 Frameworks & Segmentation

The introduction of new script languages that would outcompete PHP was a very important factor according to one informant (Part F, answer 8). PHP is a scripting language that is very easy to develop in, but also very easy to make mistakes in. According to the same informant, a big advancement came with a new generation learning to use new script languages and frameworks that don't allow for mistakes to be made in the same way. He attributes internal structure and forced parametrizing to the success of better security.

Another informant attributed segmentation as an important technical advancement that helped database security (Part C, answer 8). He stated that due to this, databases are no longer directly accessible but are instead accessed through the services that one wants to expose. Equally, another informant brings up segmentation, but in the question about what some of the threats were at the beginning of their career. He said back then there was not a lot of thought going into segmentation (Part E, answer 5), which would align with the former informants claim that it's an advancement that has helped.

4.5 Hashing and salting algorithms

Two of our informants thought that the advancement of hashing and salting algorithms was an important factor that made database security stronger. One of them talked about the fact that the security-conscious individuals used some form of hashing algorithms early on, and they knew that some of them weren't particularly strong, so they started to use them in combination with salts to increase the cost of decryption or de-hashing in the event of a security breach (Part C, answer 8). The same informant continues in that answer by once again referring to the rise of the graphics card and explains that the hashing algorithms changed very fast. Instead of using a hash and a salt, you would use an algorithm where the salt was a part of the algorithm itself, like "bcrypt", which is a standard now. Bcrypt was designed to be difficult to crack by a graphics card, which led to database leaks now not necessarily becoming mass cracked. (Part C, answer 8). If you got access to a leaked database, you would need to instead start picking your targets. Another aspect about bcrypt, according to the same informant, is that you can tweak the algorithm regarding how heavy or simple you want to make it. It lets you choose the "cost", in other words how many iterations you want to go over the encryption. This can, in theory, make it incredibly expensive to crack a bcrypt hash (Part C, answer 8). In addition, the second informant, when asked about important technical advancements, mentioned the standardized use of database-hashing, which also used salts (Part F, answer 8). This informant also states that this led to the fact that even if a database was leaked, it became way more difficult for the attacker to get anything out of it. He states that there are services that still to this day use very outdated algorithms and that they are easily crackable, this leads to those services often having unnecessarily complex password-requirements (Part F, answer 8).

Another database security topic to consider is services that have lived over a long time. In the beginning, security could have been designed to run an old hashing algorithm without a salt, but then over the years one realizes that this might not be sufficient, but one doesn't want to force old users to do a password reset. (Part C, answer 8). The informant goes on by saying that new users are set up with an updated algorithm, and later even newer users get set up with an even newer algorithm. The oldest users in the system, often the ones with the highest privileges such as administrators, are still sitting on the oldest algorithm, which can be problematic. He says that this was the case in a public Dropbox leak, where the oldest users used a very outdated hash and salt (Part C, answer 8). Considering that organizations and services perhaps don't want to force their old users to do password resets, this can according to the informant be problematic (Part C, answer 8).

4.6 Identity federation services

All informants mentioned identity federation services, with various opinions on them. Three out of four informants were positive towards federation as a security solution, with one being more hesitant about it.

When asked what the most important factor for maintaining an effective database security was, one informant answered to have intermediaries, for example to rely on someone else for authentication (Part C, answer 10).

> *"If you can rely on another party for authentication, for example, like OAuth or your google account, BankID, or something else, so that in your database you only have a pointer that says when this authentication provider gives the OK, you let the user in so that you yourself do not have to handle the entire solution. That would make things better." (Part C, answer 10)*

While the informant advises the use of federation solutions, he also admits that the downside to it is that you let a third party be responsible for your security - it's a trade-off one must think about (Part C, answer 10).

A second participant also responded with authentication when asked about the most important things to consider in database security (Part E, answer 10). He also thinks letting a third party handle it is, in many cases, a better way to go than trying to implement your own solution (Part E, answer 14).

A third participant stated he would even like to wish for legislative requirements on how to implement and get started with MFA, because it would remove a large part of the secrets in the database (Part F, answer 9). He admits that MFA can be dangerous, but he thinks we've passed a peak where we had a lot of vulnerabilities in for example OAuth 2 flows that have become fewer, the software has become more hardened and has gone through its "maiden voyage", so to speak (Part F, answer 14). In the same answer, he continues by saying that he thinks the time of 'low risk for IDPs' may now be here. He also hopes that people don't start using new frameworks, and that a monoculture or monopoly is created. He states that this can help security sometimes.

"It creates norms for how things should be. An open-source product that has reache a monopoly position can be a positive thing." (Part F, answer 14)

The fourth informant thinks that federation solutions are a bad idea. Keeping security solutions internal to your organization is important to him, the less people that have access to it the better (Part D, answer 8). At the same time, he recognizes that a full security solution developed internally from the ground up is unrealistic (Part D, answer 16). He continues by saying that you eventually need to take in parts from third parties, but he would prefer if it at least was open source so that there is an opportunity to review it. He also thinks that it's important to build a strong internal understanding for the third-party solutions you take in, and that there is a need to audit and pen-test the solutions thoroughly (Part D, answer 16).

4.7 Policies

When asked if there were any specific policies that played an important role in forming Sweden's database security, there were very mixed answers. Both to the direct question, but also to our follow-up question which asks if it would be appreciated or needed.

One participant gave a very brief 'no' when asked if they could identify any policies, and to the follow-up question also responded no (Part D, answer 9). In the same answer, he went on to say that there should be soft policies and guidelines to help organizations into the right path, but he was against mandatory policies or regulations.

Another informant wasn't sure about if there were any regulations or structured policies (Part C, answer 9). He went on to say that there are many paid solutions, but they are mostly international standards. However, he did mention that there is a law in Sweden for "protective security act", but this is a very niche regulation and doesn't affect a lot of organizations.

"Everyone has to manage as best they can on their own" (Part C, answer 9).

Equally, a third interviewee had a difficult time coming up with specific policies but mentioned compliance requirements as a form of regulation (Part E, answer 9). He goes on by saying that a compliance requirement entails a third party that reviews your application or system. He also says that it's been a lot of "companies just wanting that little check mark to show that they're done with it", which according to him invalidates the compliance requirement itself and puts more focus on the entity that actually performed the review and what type of competence that exists there (Part E, answer 9). In relation to this, the informant who was most critical against policies was also critical to guidelines/compliance requirements, stating that if an incident happens the organization can just point at and say, "but we followed the guidelines" (Part D, answer 9).

One informant gave a very direct answer to the question about policies and answered GDPR. (Part F, answer 9). In his answer he says that GDPR forces mandatory hashing, as well as having forced big companies to encrypt data to be able to lawfully send data between continents. The informant thinks that technical laws and regulations generally don't work, but says that the reason why GDPR does, is because there's a big group of people that care about it working. There are many reports, and you're forced to give users a copy of all personal data you've gathered related to them if they contact you about it, which makes it a successful law (Part F, answer 9). When asked the follow-up question, he said that there would need to be

technologies available to match the regulations, and he doesn't know what would come first; the egg or the chicken (Part F, answer 9). Continuing in the same answer, he thinks that Swedish organizations are already having a very hard time following GDPR, but fortunately enough the guidelines are incredibly clear.

The same informant continues by saying that if you want more regulations, you need clear yet generic guidelines, as it needs to work for both the small and the big firm (Part F, answer 9). In addition to this, he says that you need to measure the outcome when you write the regulation. You need to look at how you want the result to be, not at how to get there.

4.8 Swedish culture

Our informants generally thought highly about Sweden when it came to database security. Our interpretation is this may reflect a desire to be a little patriotic and a belief that Sweden is better at security, even if not having direct evidence (Part C, answer 12). A general belief of Sweden as a country being good with technology and that we are advanced compared to others was expressed (Part E, answer 12). A lot of the giant data breaches that have occurred globally were very seldom targeting any Swedish organizations, however that could also be due to the fact that the more attractive targets lie elsewhere (Part C, answer 12).

One informant believes that Swedish companies have been associated with having high security (Part F, answer 12). In the same answer, he brings up brands such as IKEA, Volvo and Ericsson, and says that he can't mention a single known or big hack against these big Swedish companies. "Risk-conservative" was a word that he used to describe Swedish engineers (Part F, answer 12). He also brought the Swedish hacking scene as the phenomenon that it was in 2007-2010 and said that the people who were dedicating themselves to criminal hacking suddenly got jobs, and today have roles as CISOs and other "fancy positions" (Part F, answer 7).

> *"During the same period when we had a hacking peak in Sweden, we also had an open-source hacking peak. We had a relatively large number of open-source developers, and a growing industry was born, so to speak. This is when Spotify made its breakthrough, and many of these other major Swedish wonders happened simultaneously, which were, after all, Swedish innovation." (Part F, answer 12).*

He continues by saying that the "Swedish wonders" had a small appetite for risk if you were to compare them with the "vulture capitalist" companies in the US, who had come and gone with what he called hockey sticks, unicorns and bitcoin companies (Part F, answer 12). In the same answer he says that there are plenty of examples of companies that have taken enormous risks, knowing that they have taken enormous risks, and have gotten caught and crashed hard because of it. For example, crypto exchanges and CEOs going to prison. He hasn't seen much of that in Europe or the Nordic countries, least of all in Sweden.

The same participant thinks that this low appetite for risk could go hand in hand with Sweden as a country not necessarily excelling in innovation when it comes to security (Part F, answer 12). According to him, the risk-taking companies that survived also revolutionized various fields with their innovation.

4.9 Recommended best practises

Two informants advocated for anomaly detection through logging of database activity. The first one explains it in the following way; previously, it has been a system account that backed up the database, now it's an individual or a personal account that, although being an administrator, triggers an alarm because it has never happened before and should not occur (Part C, answer 15). He continues by saying if you want to discover these anomalies, you need to keep a very good track of your environment. The other informant also brings this up in a similar way. He states that by logging data flows and activities, and by using this data, you can make your database security better by detecting anomalies (Part F, answer 15)

The same informant who brought up GDPR when asked about policies thinks that one of the most important things to do right now from the perspective of a Swedish organization is to map out your data flows for PII (personally identifiable information) (Part F, answer 15). If you map it out, it's clear where it is, where it has been, and that it exists. If you aren't on top of the data in this way, it becomes hard to remove it and send it out if a user asks for either of these options. (Part F, answer 15). In the same answer, he continues by saying that if you have done this mapping, it's a good first step towards pseudonymized data. In places where you have PII, you could put in a hash that would represent that person. By doing this you separate the PII in a way that you can now query the data without risking the end users PII, letting an organization be data-driven with minimized risk (Part F, answer 15). He thinks this is very important because essentially all Swedish tech companies are based on doing some type of data analysis, and they should keep doing that, but without pointing at individual users' identities (Part F, answer 15).

One of the interviewees recommends MFA, segmentation and keeping up with the evolution of coding languages, frameworks and deliberately coding applications with security in mind (Part E, answer 15).

Setting up the technology in such a way that a user cannot be deceived or phished, even if they fall for it was another answer (Part C, answer 11). The informant states that this can be done by for example removing passwords and forcing users to use tokens or solutions of a biometric nature.

> *"And then, of course, there are the boring and normal precautions that have always helped; do not grant higher privileges than necessary, one should not be an administrator on the computer they work on, but rather on another device etc. There is a lot to consider. But if you start with strong authentication, hardened clients, and good detection capabilities on the clients, you will come a long way." (Part C, answer 15).*

5 Discussion

5.1 Data breaches

5.1.1 Cost of data breaches

Our empirical results highlighted a consensus among the informants that Sweden as a country holds a high standard when it comes to preventing data breaches. The theoretical results that we could find were very limited if scoped to academia. There are reports from the industrial sector that analyze the cost of data breaches, and we could see that Sweden ranks among the lowest security-cost in the world (IBM Security, 2022). However, using this as the only metric will not lead to any relevant or good conclusions, as the good ranking in data breach cost could be due to the fact that perpetrators don't see any value in targeting Swedish organizations (Part C, answer 12). Another possibility is that the breaches in Sweden have been underreported and that the IBM study hasn't captured some of the more well-known breaches. It is probably worthwhile to conduct an in-depth study of the cost of data breaches in Sweden as the IBM report is very wide and global based on a representative sample of under 600 cases (IBM Security, 2022).

Another interesting finding in our results is that of the Swedish hacking phenomena during 2007-2010. Many people who were dedicating themselves to criminal hacking left the scene and instead obtained "white hat" employment, where they today boast roles as CISOs and other 'senior' level positions (Part F, answer 7). Essentially, many of the people causing the data breaches during the time switched sides and instead started focusing on bettering the security. This could also play a role in why the cost of a data breach is low in Sweden. Further exploration could investigate how this transition has shaped the security landscape in Sweden, and whether this practice is more prevalent in Sweden than in other countries.

There are several factors that contribute to how much a data breach will cost for an organization. The cost is often measured in monetary value, but it's important to keep in mind that there is more to the total cost picture than the actual fines and ransoms. The U.S. Federal Trade Commision (2021) outlines how a direct expense can look like after a data breach. Our empirical results didn't provide us any insight into specifics when it came to fines and numbers in that sense, however it did provide results that are in line with the rest of the theory. "Gunneboläckan" is a Swedish data breach that falls under many of the consequential categories that were brought up in theory (Part F, answer 7). As a company that specializes in security, perhaps most importantly bank vaults and physical security solutions for prisons, it becomes a national security concern when all of the drawings and blueprints for Gunnebos products lie open and are leaked publicly on the internet (Part F, answer 7).

5.1.2 Data breaches from an information security perspective

Depending on the extent of the data breach, and the organizational variables in the breached content, the breach can affect all three aspects of the CIA-triad. Our empirical results pointed to a lot of data breaches leading to a public leak of organizational data, such were the cases for Gunneboläckan, the Swedish Transport Agency and the Swedish Tax Agency (Ryberg, 2013; Chirgwin, 2017; Mårtensson et al., 2017; Part F, answer 7; Part D, answer 7; Part C

answer 7). Looking at this from the theoretical perspective, the leaks fall under the category of unauthorized disclosure of confidential information, which is a loss of confidentiality (Elmasri & Navathe, 2015, p.1123). The theory suggests that failing to keep your data confidential could result in "loss of public confidence, embarrassment, or legal action" (Elmasri & Navathe, 2015, p.1123). This was exemplified further in the theoretical background with real-life examples of fines (The U.S. Federal Trade Commission, 2021). Furthermore, the theory also suggested that loss of confidence and reputation "can have a profound and lasting impact on customer attrition and competitive advantage" (IBM, 2014). A Swedish example of this loss of public confidence, was the Logica hack in 2010-2012, in which the appellant court wrote in its verdict that the damage to trust suffered by the Swedish authorities due to the breach had been great (Hovrätt, Dom B 6402-13). Our empirical results also found there was a loss of trust towards the Swedish authorities after the mishandling of databases from the Swedish Transportation Agency (Part D, answer 7).

Our empirical results also show that data breaches can lead to the loss of integrity. When trying to restore database backups after incidents, it's common that the data gets back corrupt, rendering it useless (Part F, answer 10).

The theory suggests the "original" version of the CIA triad to still be an industry standard (Van Der Ham, 2021; Lundgren & Möller, 2017; von Solms & van Niekverk, 2013). However, when discussing with the experts about generic security practices for databases, a lot of the focus centered on accountability and authentication (Part C, answer 10; Part E, answer 10, 12, 15). These are the two additional key terms that get added to the triad in its criticism (Stallings & Brown, 2018). Two out of the four experts recommended logging of database activity as a best practice for database security so that you can detect anomalies and keep track of your environment (Part C, answer 15; Part F, answer 15). This practice adheres with the accountability concept of the "extended" triad. Three out of four experts also mention robust user authentication as a best practice (Part C, answer 15; Part E, answer 10; Part F, answer 9). The thoughts and opinions from the industry experts seem to adhere to the "outcome" that the CIA criticism is trying to convey, even if they don't specifically mention the CIA triad, its concepts and outcome-oriented goals are in line with the story the informants are telling.

5.2 Historical evolution of database security in Sweden

In examining the past 20 years, we can get a rich understanding of threats, solutions and policies that have emerged, offering insights that can help better understand the field and guide future efforts.

5.2.1 Threats

The empirical results show that at the beginning of the 2000s, threats to database security were primarily characterized by SQL-injections and inappropriate storage of credentials in plaintext. Direct concatenation of user data with database data before queries was often allowed, as detailed by our informants (Part E, answer 5; Part F, answer 8). The theory also describes SQL-injections as a problem area, although mainly when analyzing the 90s (Lesov, 2008). Lesov (2008) states that the research community often contemplates concerns well in advance of them being tackled by actual implementations. It is therefore a possibility that the research on SQL injections and the experience from our informants of when they were

relevant as a threat risk for organizations don't line up entirely. As SQL injections became more recognized, the developer community responded by adapting new frameworks that enforce parametrization of SQL queries, effectively mitigating the threat (Kindy & Pathan, (2011); Part F, answer 6; Part E, answer 7). Regarding the inappropriate storage of credentials, the informants didn't delve deeper into the subject, but implicitly voiced that it was a moronic practice.

However, the threats have since evolved. The underground hacking movement in Sweden from 2007 to 2010 marked an important shift, and after this point in time "hacking for the sake of hacking" didn't exist to the same extent (Part F, answer 7). Criminal hacking became more organized, with the rise of "initial access brokers", which are threat actors that sell stolen databases or accesses (Part F, answer 10). Our results also showcase that state-sponsored groups became more relevant after this time period and that they are behind a major part of "hacks" today (Part F, answer 10; Part E, answer 6). Modern gaming graphics cards also further complicated threats against database security, as they enabled password cracking with an efficiency unlike anything seen before (see: 2.9.4 Hashing and Salting; Part C, answer 8; Part F, answer 6 and 8).

5.2.2 Solutions

The advancements of solutions to some of the discussed threats faced in the field of data security have been characterized by technical advancements and strategic shifts in best practices. One of the most significant advancements, as highlighted in our empirical results, was the implementation and evolution of hashing and salting algorithms (Part C, answer 8; Part F, answer 8; see: 2.9.4 Hashing and Salting).

In the 2000s, there were security-conscious individuals who recognized that certain hashing algorithms were not sufficiently robust. This led them to combine the algorithms with salts to increase computational cost of decryption and de-hashing in the event of a data breach. However, the addition of gaming graphics cards necessitated further advancements of algorithms. The response was the implementation of algorithms like bcrypt, which is an algorithm where the salt is an integral part of the algorithm itself. Bcrypt is an algorithm particularly difficult to crack by graphic card (Part C, answer 8; see: 2.9.4 Hashing and Salting). This added an extra layer of security in the event of a data breach. Modern day algorithms are also flexible, letting you tweak the computational cost in the configuration (Moriarty et al. (2017); Part C, answer 8; Part F, answer 8). As a result, it made it significantly more difficult to attack databases en masse, and you needed to start picking your targets (Part C, answer 8; see: 2.9.4 Hashing and Salting). In addition, the theory seems to follow the same phenomena of processing a topic prior to it becoming relevant to industry practitioners. Most of the algorithms mentioned by our informants were developed in the 90s (Rivest, 1992; National Bureau of Standards, 1995; Dobbertin, Bosselaers & Preneel, 1996; Provos & Mazieres, 1999). However, the informants recalled algorithms being introduced at a comparatively later stage in history as a direct response to the organizational consequences of data breaches. Again, exemplifying how the research community often contemplates concerns well in advance of them being tackled by actual implementations.

In addition to the advancements of hashing and salting algorithms, the shift from languages like PHP towards newer languages and frameworks that enforce better security protocols also played an important role in the advancement of database security (Part F, answer 8). The newer frameworks that, on a technical level, simply didn't let you make the same type of

mistakes, started replacing the old ones. It's worth noting that some services still run PHP to this day, and that our empirical results state that leaks are a "daily occurrence" among these services (Part F, answer 8).

5.2.3 Policies

The empirical results show that the opinions regarding policy-related questions among the informants varied. Some informants couldn't identify any specific regulations or structured policies that they thought have made a substantial impact in the field (Part D, answer 9; Part C, answer 9). This was a surprising finding in that the informants are recognized experts in the field. Perhaps a further study can better understand why policies and regulations are out of line with expert opinion. The suggestion is to have a public policy process that is well-coordinated and agreed upon by noted industry experts.

Our expert with the most experience in the private sector voiced the opinion that there should be less regulation in general, and that the market should regulate itself (Part D, answer 9). By following this approach, customers would naturally seek themselves to the organizations that they deem most secure. However, the experts with a lot of experience working for the state security services, mostly the police and security police, thought that regulations could be of benefit, even if, pessimistically stated, due to the track record so far not having been great (Part C, answer 9; Part E, answer 9). The consensus seemed to be that there were always good intentions behind the regulations, but that they often failed to produce results, and that if it were possible to implement regulations that produced satisfactory results, it would be welcomed. The expert whose answer stood out the most, explained how GDPR was a regulation that has shown very good results so far (Part F, answer 9). This, according to him, is based on the fact that GDPR is outcome oriented. His opinion was that other regulations of the prescriptive nature led to companies "ticking a box" just to have it ticked, and not establishing a genuine security solution. The theory also suggests that GDPR is outcome oriented, and that it entails a holistic view of the security process, instead of a "tick-box compliance exercise" (Mansfield-Devine, 2017). For the sake of the industry, we hope to see more of this approach as it seems to solve a lot of the issues regarding regulations brought up by the experts.

5.3 Suggested Best Practices

Much of the academic literature we found specifically discussing best practices for database security was outdated either as aggregated results of older findings or the findings themselves were from 10 years ago, or older. Some of the practices mentioned in the literature were also mentioned in our empirical data, often in a historical context (Part C, answer 5, 6, 7; Part D, answer 6; Part E, answer 5, 15; Part F, answer 6). There were several database security best practice recommendations made by our interviewed experts that were not, or at least very seldom, mentioned in the literature. This section outlines some of the recommendations we found interesting based on the empirical data gathered.

Two-factor authentication or MFA was mentioned by three experts as both being more readily accessible today as well as Sweden being at the forefront of security applications such as BankID for securing authentication to web services. Furthermore, the use of MFA appears like a good security measure to secure authentication and user authorization, e.g., by not having to trust a user to resist a targeted attack by not giving out their password (Part C, answer

11, 14; Part E, answer 12; Part F, answer 9, 14). An expert also noted that because of the future prevalence of MFA that threat actors will instead focus on stealing sessions instead of credentials in order to gain and sell access to systems (Part F, answer 14).

All experts suggest the importance of good coding practices, such as security conscious coding, code reviews and threat modeling during development, and tools such as frameworks that had security defaults as well as automated testing and to help find vulnerabilities (Part C, answer 10; Part D, answer 8, 10, 12, 16; Part E, answer 6, 8, 10, 15). Two experts argue for the importance of Zero Trust and that it can, and that it should be used to secure access to databases (Part D, answer 8, 10, 12; Part F, answer 10).

There is no real consensus on whether federation services (see: 2.9.5 Centralized & Federated Identity Management Systems) are a positive or a negative force for database security. One expert is of an opinion that federation can pose a risk as there would be two IdP services that holds the users' sessions, i.e., two points of attack and potential data leakage, but that the development of the technology and standards is moving forwards and security holes are being addressed and mentions an IdP that seems robust, Keycloak, that also has a large userbase (Part F, answer 14). Another expert cites security concerns with outsourcing authentication and authorization to third parties outside one's organization (Part D, answer 8, 16). One expert says it's hard to answer anything general about federation but that their gut feeling is that it's better to trust a third party with authentication than implementing your own solution (Part E, answer 14). This logic makes some sense in that global service providers like Microsoft would have more resources for security than most of the small-to-medium business. One expert cites outsourcing one's authentication to a third party is one of the most important factors for effective database security (Part C, answer 10) demonstrating that there is no real consensus on the matter.

One expert suggests an interesting approach towards securing data inside databases in an encrypted form whilst still letting queries search the database for information without the database data requiring to be decrypted on the database server or whilst the data is in transit. This is referred to as Homomorphic encryption and is being actively developed (see 2.9.1 Encryption; Part F, answer 9). Homomorphic encryption also allows for homomorphic encrypted data to be merged with other data allowing for big datasets to travel regulatory borders such as the GDPR zone whilst still having the data encrypted but "usable" for analysis.

Anomaly detection though SIEM or other systems would allow defenders to view attacks that otherwise might go undetected though aggregating data from different sources. Anomaly detection can also be extended to user authentication through analyzing signals such as browser, device, or geo-location (Part C, answer 11, 14, 15; Part F, answer 16).

All experts, in different ways of expressing it, state that there is no single "silver bullet" that solves the challenges of database security. Instead, they suggest approaching the problem with a holistic approach involving a myriad of mostly technological methodologies, from old tried and tested best practices, such as firewalls, network segmentation, hashing and salting and access controls, to utilizing newer approaches mostly focused on authentication, authorization and anomaly detection and new encryption schemes (Part C, answer 10, 15; Part D, 8, 10; Part E, answer 10, 15; Part F, answer 10, 14, 15).

6 Conclusion

Database security is a "must-have" for a modern society to operate. Like any subject, one needs to understand the past in order to better prepare for the future. The aim of this thesis is to place database security in a historical context because, in our view, there is a significant gap of historical knowledge missing in literature and research on this topic, especially for Sweden. We wanted to directly hear from notable experts in the field regarding the "story" of database security, and better understand the critical components of database security that need to be addressed. We also conducted an exhaustive search for any literature and research regarding database security with a holistic and historical context.

During our extensive research we found that very little research had been produced concerning the historical context of database security internationally, and even more so in a Swedish context to the extent that we did not find any Swedish sources that deal with this subject. We believe that the basis for our field's lack of knowledge in this area is the result of a combination of a very high degree of specialization resulting in "too many trees and not enough forest", and a very fast-moving industry where historical context is often overlooked and "just move things along".

We relied heavily on our in-depth interviews with experts in the cyber security field in Sweden to obtain a more granular understanding of the historical context of database security. Using in-depth qualitative interviews provides a rich source of historical context and future direction. Experts, all having around 20 years of experience in the field, that can see the entirety are few. Their feedback seems to align with our premise that our industry needs to view database security in a more holistic way. This is a common theme across all interviewees. Another interesting takeaway is that most experts shared suspicions in the usefulness of information security policies in protecting databases and instead wanted to rely on technical countermeasures.

Our key takeaways from this effort are:

- The economic and social costs of security breaches are going up, not down. Database security is becoming more complex, not less. This implies a need for the security community to develop much more holistic security solutions in order to reduce risk.

- Database security and IT security in general is a highly specialized field bifurcating the industry. An effort to create more "generalized specialists" who can provide holistic and cross-specialty overviews could go a long way to improving security and reducing risks.

- There is on-going debate in the data security community whether to centralize or decentralize data security. The private sector prefers the latter, the public sector the former. In our view, both directions are needed, and the steering principle should be whether the data that needs protection is of national and public interest or not.

- The "history book" of database security, particularly for Sweden, has still to be written. Bridging this knowledge gap could help better prepare for future cybercrimes.

Appendix

Part A: Interview Guide

The interview guide is presented here in both in English and in Swedish (italicized)

Pre-interview information

If the interview takes place physically, via video link or voice call, the audio will be recorded from this. This recording will first be transcribed into text and then destroyed as soon as the transcription is completed. All collected material collected from the interview will be used only for the purpose of research and publication of a BSc thesis.
Om intervjun sker fysiskt, via videolänk eller röstsamtal så kommer ljudet att spelas in från detta. Denna inspelning kommer först att transkriberas till text för att sedan destrueras så fort transkriberingen är avklarad. All insamlat material som samlas in från intervjun kommer endast att användas för forskningsändamålet och publicering av en kandidatuppsats.

We would like to include information about the candidate in the response appendix such as: **Name, job title, company/organization**, date of interview, **place of interview** and method of interview.
Vi vill gärna ha med information om kandidaten i svarsappendix såsom:
***Namn**, **arbetstitel**, **företag/organisation**, datum för intervjun, **plats för intervjun** samt metod för intervju.*

If the candidate wishes, we can anonymize one or more of the parts in bold about them in the answers. The candidate can also choose to cancel the interview at any time or withdraw any part/parts or all of the interview.
Om kandidaten önskar kan vi anonymisera en eller flera av de fetstilade delarna om den i svaren. Kandidaten kan också välja att avbryta intervjun när som helst eller dra tillbaka någon del/delar eller hela intervjun.

The candidate can choose to answer the questions in English or Swedish.
Kandidaten kan välja att svara på frågorna på engelska eller svenska.

The candidate will be notified upon completion of the study that the essay can be sent to him if he wishes.
Kandidaten kommer att meddelas vid färdigställd studie att uppsatsen kan skickas till denne om den vill.

Does the candidate agree to this?
Godkänner kandidaten detta?

Interview questions in English and in Swedish (italicized)

1. Name, job title and organization?
   Namn, arbetstitel och organisation?

2. For how long have you been active in said organization?
   Hur länge har du varit verksam inom denna organisation?

3. What are your current responsibilities?
   Vad har du för ansvarsområden i dagsläget?

4. For how long have you been active in the field of cyber security in Sweden?
   Hur länge har du varit verksam inom området cybersäkerhet i Sverige?

5. What were some of the most common database security threats when you began your career?
   Vad var de vanligaste säkerhetshoten mot databaser när du började din karriär?

6. How has the landscape of database security threats changed in Sweden since then?
   Hur har landskapet för säkerhetshot mot databaser förändrats i Sverige under tiden du varit verksam?

7. What have been some of the most significant database security incidents in Sweden, and what lessons were learned from those experiences?
   Vilka har varit några av de mest betydande incidenterna gällande databassäkerhet i Sverige, och vilka lärdomar drogs av dessa erfarenheter?

8. What were the key technological advancements or practices that contributed to the evolution of database security in Sweden?
   Vilka var de viktigaste tekniska framstegen eller metoderna som bidrog till utvecklingen av databassäkerhet i Sverige?

9. Can you identify any specific regulations or policies that have played a critical role in shaping database security in Sweden?
   Kan du identifiera några specifika regler eller policyer som har spelat en avgörande roll för att forma databassäkerheten i Sverige?

   Följdfråga: tycker du det behövs?

10. In your opinion, what are the most important factors for maintaining effective database security in the context of Swedish information systems?

Vilka är enligt dig de viktigaste faktorerna för att upprätthålla en effektiv databassäkerhet i kontext av svenska informationssystem?

11. How have Swedish organizations typically approached database security? Has this approach evolved over time? If so, how?
Hur har svenska organisationer vanligtvis ställt sig till databassäkerhet? Har detta tillvägagångssätt utvecklats över tiden? Om så är fallet, hur?

12. Are there any unique aspects of Swedish culture or society that have influenced the way database security is handled in the country?
Finns det några unika aspekter av svensk kultur eller samhälle som har påverkat hur databassäkerhet hanteras i landet?

13. How do Swedish database security practices compare to those in other countries? Are there any specific strengths or weaknesses that stand out?
Hur står sig svensk databassäkerhetspraxis jämfört med den i andra länder? Finns det några specifika styrkor eller svagheter som sticker ut?

14. What emerging technologies or trends do you think could have the most significant impact on database security in the coming years?
Vilka nya teknologier eller trender tror du kan ha den största inverkan på databassäkerheten under de kommande åren?

    Följdfråga: tycker du federering är bra eller dåligt, vill du se mer eller mindre av det.

15. Based on your experience, what would you recommend as best practices for organizations in Sweden to enhance their database security moving forward?
Baserat på din erfarenhet, vad skulle du rekommendera som bästa praxis för organisationer i Sverige för att förbättra sin databassäkerhet framåt?

16. Anything else you would like to add?
Finns det något annat som du vill tillägga?

Part B: Interview Definition of Terms

Acronyms and terms	Definition and explanation
SDK	Software Development Kit
OpenCL	Open Computer Language, framework for writing programs on Central Processing Units (CPUs), GPUs and FPGAs
CUDA	Compute Unified Device Architecture, parallel computing API that allows software to use GPUs for general purpose processing
WPA	Wi-Fi Protected Access, a standard of security mechanisms for wireless networks, WPA passphrase hashes collected are feasible to crack offline
FPGA	Field-Programmable Gate Arrays, an integrated circuit designed configurable by a customer or a designer after manufacturing
VHDL	VHSIC Hardware Description Language, can model the behavior and structure of digital systems at multiple levels of abstraction
AGSM	Adaptive Generalized Spatial Modulation, used in MIMO wireless system transmission technology
PHP	PHP is a general-purpose scripting language geared toward web development
IP	Internet Protocol, used for network layer communications protocol in the Internet protocol suite for relaying datagrams across network boundaries
LAMP	Linux, Apache, MySQL, PHP/Perl/Python is a software stack for web applications
Nginx	An open-source web server that can also used as a load balancer or reverse proxy
MD5	MD5 message-digest algorithm (hash function)
SHA1	Secure Hash Algorithm 1 (hash function)
GPU	Graphics Processing Unit (graphics card)
bcrypt	bcrypt is a password-hashing function
BankID	BankID is an electronic identification system in Sweden.
SIEM	Security Information and Event Management provide real-time analysis of security alerts from applications

WAF	Web Application Firewall, filters, monitors, and blocks HTTP traffic
EC2	Amazon Elastic Compute Cloud, allows users t rent virtual computers
LLM	Large Language Model
Regex	Regular expression, shortened regex, sequence of characters that specifies a match pattern in text
VLAN	Virtual Local Area Network, a broadcast domain that is segmented and isolated in a network
SQL	Structured Query Language, used to manage data held in relational databases
IEC	International Electrotechnical Commission
ISO	International Standards Organization
SÄPO	SÄkerhets POlisen (Swedish Security Service)
JWT tokens	JSON Web Toke, a web standard allowing for signature and/or optional encryption whose payload holds JSON
Nis / NIS2	European security regulations, Network & Information Systems regulation aimed at rising levels of cyber security and resilience of state and public information systems.
gRPC	gPRC Remote Procedure Calls, open-source cross platform high performance Remote Procedure Call (RPC) framework

Part C: Interview Lars Otterskog – Swedish Police Authority

Interview participant: Lars Otterskog
Title: Cyber Security Specialist
Organization: Swedish Police Authority
Past experience: Cyber Security Specialist at SÄPO, Cyber Security Specialist at the Swedish Police Authority, Cyber Security Consultant at Certezza AB, Presales Technical Consultant at BorderWare Technologies, Senior Technical Support Specialist at Symantec
Date: April 19th, 2023
Method: Teams meeting interview
Length: 47:30 min

Q	Answers & Follow up Questions	Code
1	***Namn, arbetstitel och organisation?***	
1	Precis. Det är Lars Otterskog, arbetstitel är väl säkerhetsspecialist, och organisation är på Polismyndigheten.	
2	***Hur länge har du varit verksam inom denna organisation?***	
2	Vad blir det nu.. 13 år blir det, först polismyndigheten, sen säkerhetspolisen och nu är jag tillbaka på polismyndigheten.	
3	***Vad har du för ansvarsområden i dagsläget?***	
3	Det är ganska brett men det är dels IT-säkerhetsarkitektur, det vill säga när vi tar fram nya nät, nya produkter och liknande så har jag säkerhetssyn på det hela. Det är också att köra redteamövningar mot befintlig infra och testa våra SOC samt att stötta verksamheten med utredning när det är något som en IT-utredare kanske inte har koll på, eller det drar mer åt säkerhetshållet än traditionell forensik, då brukar jag hjälpa till med den typen av ärenden. Så det är hyfsat brett där men det som är mest relevant då för det här är väl mer IT-säkerhetsarkitektur då i design.	
4	***Hur länge har du varit verksam inom området cybersäkerhet i Sverige?***	
4	Det är nog sedan.. jag skulle säga 2005	
5	***Vad var de vanligaste säkerhetshoten mot databaser när du började din karriär?***	
5	Tittar man lång tid tillbaka så var det precis när den här fasta uppkopplingen började bli normal i Sverige. Dvs att man både som privatperson och företag fick sin uppkoppling, och då skulle ju alla sätta upp sina egna websidor och sina egna IT-system. Det man gjorde då helt enkelt var att ladda ner en bundling av apache, mysql och php, så kallade lam eller amp paketet. Och då, det innebar ju att databaser var exponerade direkt mot internet. Det fanns liksom inget tänk med segmentering, front end, back end och brandväggar. Utan man hade en IP-adress och det var där man körde alla sina tjänster. Så på den tiden kunde det snarare vara ett problem med dåliga lösenord till databaser, eller	MC HIS

	defaultlösenord tillochmed till databaser. Så där fanns det ju inget säkerhetstänk alls då i början. Det var ju problemen då skulle jag säga, om man skulle ta de största problemen.	
5	***Följdfråga: Nu när vi pratar lamp och nginx typen vad den nu hette.. Du pratade om default portar, så de låg öppna i brandväggen utåt? Och så hade de default credentials till dem systemen?***	
5	Ja precis, det var inte ens på tal om brandväggar då, iochmed det var såpass nytt det här med fasta uppkopplingar. Man hade ju ett gäng IP-adresser och hade man en server kopplade man in den i sin router. Den fick en publik IP-adress rakt ut på internet. Det var frikostigt på den tiden, det fanns hur många som helst. Så då låg den exponerat istället för brandväggen som man har idag. Så ville någon komma in på den här sidan så var det bara att koppla upp mot databasen på samma IP-adress som webtjänsten hade. Där låg det också ftp och allting som man behövde för att drifta den här tjänsten. Det var så illa det såg ut i början.	HIS
6	***Hur har landskapet för säkerhetshot mot databaser förändrats i Sverige under tiden du varit verksam?***	
6	Det har förändrats på sättet att numera har man ju ett visst tänk när man sätter upp webtjänster eller tjänster överlag på internet. Man har ju tack vare att det blir en brist på IP-adresser så har man en brandvägg, och då får man öppna upp portar mot sin tjänst, och sen så blir det att den tjänsten kopplar upp mot en databas i backend. Så att även om det inte är något medvetet val som en person gör så blir det så av design bara för att man inte kan köra publika IP-adresser överallt. Vilket har lett till att numera om man vill komma åt informationen i en databas så måste man då först traversera den tjänst som den är ämnad att serva, tex en webbtjänst eller någon app eller liknande. Så man har ju flyttat problemen då från nu är det inte att databasen är det sårbara, utan den blir det via den tjänst som den är där för att serva. Så det är väl så det har förändrats nu då kanske skulle jag säga.	HIS
7	***Vilka har varit några av de mest betydande incidenterna gällande databassäkerhet i Sverige, och vilka lärdomar drogs av dessa erfarenheter?***	
7	Ja precis, nu har jag en wikipediasida uppe här nu så jag inte säger fel eller för mycket. Men jag skulle säga det är väl egentligen två saker som sticker ut. Det ena är väl att åtkomsten till databaserna hos skatteverket, där när Gottfrid Svartholm Warg hackade sig in och började läcka personuppgifter på folk med sekretess. Det är väl en sån där sak. Men jag tror att Jesper har bättre koll på just den biten. Den andra stora det är väl, som varit medialt stort, är när personal utomlands fick åtkomst till transportstyrelsens databaser. Och det är väl också en stor händelse i svensk historia, och där var det då att man gav åtkomst till någonting som inte skulle ha givits, och att man inte säkerhetsprovade personalen. Det tyder också på att man kan ha otroligt många skyddsåtgärder på plats, tekniskt sådana, men om det blir några fel i form av tilldelning av behörigheter eller liknande så slås ju allt sånt här ut. Så säkerhet behöver inte bara vara av teknisk natur, utan det kan vara av processer och rutiner också som man ska ta hänsyn till när det väl kommer till informationsåtkomst	HIS

8	***Vilka var de viktigaste tekniska framstegen eller metoderna som bidrog till utvecklingen av databassäkerhet i Sverige?***	
8	En av de stora sakerna var att man blev tvingad att börja segmentera upp sina system. Databaser ligger inte längre åtkomliga direkt utan från de tjänster som man vill exponera. En annan som tvingade fram någon form av bättre säkerhet är att de säkerhetsmedvetna i början använde någon form av algoritmer och saltningar, man kände på sig att en MD5a eller en SHA1 är inte säkert, därför använder man de i kombination med en salt för att öka på kostnaden om man vill vid ett intrång dekryptera eller dehasha någons lösenord. I samband med det, när grafikkort och GPU blev åtkomliga, som tidigare var specialdesignade kretsar som gemene man inte kunde göra. Inte ens företag utan det var ju statnivå på det hela. Helt plötsligt kunde vilken gamer som helst använda sitt grafikkort och få ut bättre prestanda än specialdesignade chip kunde göra. Det ledde ju till att hashning algoritmerna förändras väldigt fort, vad som ansågs vara säkert. På senare tid har det blivit att istället för att man använder någonting plus ett salt, så använder man en algoritm där saltet är en del av själva algoritmen, tex bcrypt som är en standard. Den är designad för att vara svårforcerad med grafikkort på grund av den minnesbandsbredd som behövs för att hantera den. Det har också blivit en förbättring på det sättet då, man kan inte längre massknäcka databaser som blivit läckta. Med bcrypt kan man också ställa in någonting som kallas för kostnad, dvs hur många iterationer ska man göra av någonting. Man kan ju ställa upp det här otroligt långt, om man vet att man kan acceptera att en användare som loggar in på sin tjänst, det är ok att de får vänta en sekund vilket är en orimligt lång tid, men vill man vara säker kan man säga att varje inloggning kommer ta en sekund med sin tjänst. Detta gör att när jag då ska dekryptera någonting så blir det otroligt kostsamt för mig att göra det. Där kan man också tweaka i algoritmerna hur pass tungt eller hur pass lätt man vill göra det. Det skulle jag säga var en stor förändring, att en databasläcka inte längre kan bli massknäckt utan att det kostar så pass mycket att man väljer ut sina mål istället.	HS
8	***Följdfråga: händer det fortfarande idag att det sker massläckor på datbaser och det visar sig att organisationer helt enkelt inte använt sig av dessa best practise enkryptionerna?***	
8	Ja precis, det här är ju också ett roligt exempel påd et hela, att om jag tar fram en tjänst som lever över tid, i början kanske den designades för att köra MD5 utan salt, sen kommer man på att det här är inte bra men man vill inte tvinga användarna att göra lösenordsåterställning. Nya användare får SHA1, sen lite senare med tiden kommer man på att det här var inte heller bra så ytterligare får bcrypt, vilket betyder att de äldsta användarna har en algoritm som är väldigt lätt forcerad, medans de senast skapade användarna har en väldigt säker algoritm. Detta kan ju vara lite problematiskt, det har varit några sådana exempel i publika läckor som Dropbox till exempel. Där sattes användare från början upp med saltade SHA1, varpå senare användare fick bcrypt.	HS HIS
8	***Följdfråga: vi antar då att de första användarna som skapas i en databas kanske är de som har högre behörighet som administratörer osv. Ser du då att de ligger kvar på gamla opålitliga hashar?***	

8	Ja precis, de första då tillexempel som du säger admin, användare #1, #2 #3 osv, root och admin konton. Det känns rimligt att de skulle ligga kvar då på det ursprungliga om man inte då ser om sitt eget hus och roterar sitt lösenord bara för att hamna i den här nya snurran. Det är ett problem.	
9	***Kan du identifiera några specifika regler eller policyer som har spelat en avgörande roll för att forma databassäkerheten i Sverige?***	
9	Policy.. jag vet inte om det finns några uppstyrda eller strukturerade policys som hanterar det. Det finns då betalningslösningar, men det är inte så mycket för sverige utan internationella standarder som säger att man tex inte får lagra kreditkortsuppgifter i dina databaser utan det ska ske hos någon annan betalningsleverantör. Sånna saker är ju positivt men i övrigt när det kommer till design och typer av hashningar och liknande är det tyvärr oreglerat vad jag känner till. Det finns vissa när det kommer till säkerhetsskyddslagen där det står hur man ska segmentera sina nät och tjänster, och därtill kan man läsa då att hanterar man säkerhetskänslig verksamhet ska man göra på ett visst sätt, även om det uttryckligen inte säger att det databaserna du ska segmentera. Men det blir så att om man ska segmentera alla tjänster så blir det också databaserna, men detta är verkligen nichefall, det är inget som slår på den stora massan. Alla får klara sig så bäst de kan själva.	CUL
9	***Följdfråga: skulle du som sitter i en brottsbekämpande myndighet vilja se någon form av nationella råd till den offentliga och privata sfären i Sverige angående hur de borde konfigurera sina databaser osv?***	
9	Det finns ju vägledning för säkerhetsskydd, men i 99.9% av fallen av andra offentliga verksamheter och företag har ju ingen konkret vägledning att gå på. Det är det som behövs, det finns ju väldigt mycket prat om ISO27001 och abstrakta saker, men när det väl kommer till kritan; när man ska bygga någonting, hur gör man då? Det är den typen av vägledning som behövs och någonting man kan gå efter. Som mycket annat är det bra om det fanns.	ISO
10	***Vilka är enligt dig de viktigaste faktorerna för att upprätthålla en effektiv databassäkerhet i kontext av svenska informationssystem?***	
10	Att försöka ha någon form av förmedlare eller någonting. Det är väl lite så att man inte ska designa sina egna krypton och egna lösningar. Kan man förlita sig på någon annan part för autentisering tillexempel, som oauth eller sitt googlekonto, bankid eller någonting annat, så att i din databas har du bara en pekare som säger att när den här autentiseringsförmedlaren säger ok släpper du in användaren så att du själv inte måste hantera hela den här lösningen. Det skulle göra saker och ting bättre. Nackdelen är att du förlitar dig på en tredjepart för autentisering, gör de fel kommer de komma in hos dig med då. Det är en avvägning man måste göra, känner man att man kan designa en lösning som håller över tid och att man har resurser att säkerhetsgranska den innan driftsättning, göra ett pentest av den, kodgranska osv, så är ju det en fördel också.	MFA ZT FED
10	***Följdfråga: Du nämnde Bankid, skulle du vilja se större användning av Bankid hos företag osv då?***	

10	Där får man också titta lite på kontexten. I det stora offentliga så har ju bankid blivit någon form av standard, på gott och ont iochmed att det inte finns något statligt alternativ så har det blivit det bästa man kan åstadkomma. Inom företagsvärlden kanske det är vanligare då att man har tex microsoftkonto, googlekonto eller motsvarande som passar den typen av verksamhetssfär. Plus att allting behöver inte vara bundet till ett personnummer eller individ, utan det kan vara något mer abstrakt då vilket möjliggörs med de andra parterna då. När det kommer till databassäkerhet är det ofta så att någon administratör blir lurad, eller någons cookie kommer på vift, eller lösenord eller gitkonto eller sådär. Det behöver inte vara ett intrång mot tjänsten, utan mot någon med höga behörigheter i en tjänst och därgenom tar en backup av sin databas och kopierar ut den. Det är ofta välidgt tacksamt att göra det via det sättet för man har ett gränssnitt som är gjort för att tanka hem den här stora datamängden. Man behöver inte hålla på med hacktools, reverse shell för att försöka exfiltrera från en tjänst för att få ut datat, utan man använder den metodik som är ämnad och anpassad för det här. Det är väl två delar då; se till att ingen hackar sig in på tjänsten eller hittar fel i din autentiseringslösning. Alternativt se till att ingen kommer över en administratörs behörigheter för de här tjänsterna. Iochmed att saker och ting blir mer härdat, det tas fram frameworks, man behöver inte göra så mycket själv, det sker mer federeringsautentiseringar. Då kanske det är lättare att phisha någon administratör.	MFA FED
11	***Hur har svenska organisationer vanligtvis ställt sig till databassäkerhet? Har detta tillvägagångssätt utvecklats över tiden? Om så är fallet, hur?***	
11	Det sker ju en förändring, eller det har skett en förändring över tid att det blir bättre allting. Samtidigt så tar det ju tid, synen på hur man ska skydda sin information har ju också förändrats, och det har varit väldigt mycket brandväggar, säkerhetslösningar och produkter som köps in för att skydda mot angrepp, dvs ids, ips, antivirus och liknande. Men det känns nu som att fokus skiftar mer till att försöka härda och identifiera, och det är väl vägen framåt som jag ser. Att inte försöka förhindra, eller hitta attackmönster som ändå kommer förändras vid varje attack. Så att använda såhär SIEM lösningar för att aggregera dataposter och försöka hitta konstigt beteende. Det är väl där den stora förändringen sker, man kan helt enkelt inte förlita sig på en WAF tex att den ska plocka en SQL-injection, det kommer vara så många andra attackvägar, så man får helt enkelt se att oj det här informationssystemet försökte ansluta till ett annat informationssystem som det aldrig försökt ansulta till förut tex, nu har en användare loggat på det här systemet, den har aldrig loggat på tidigare. Att försöka hitta sånna här typer av logposter då. Det är ju någon förändring som har skett på senare år, väldigt i närtid ska jag säga. Det har varit väldigt varierande nivå där, hur pass snabbt man anpassar sig till den nya verkligheten.	HIS FT AM
11	***Följdfråga: det är väldigt många tekniska lösningar som kommit på senaste, om man tittar på den "mjuka" delen, eller mänskligt fokuserade, social engineering osv. Hur har man ställt sig till det i Sverige och har det utvecklats någonting på senaste år?***	
11	Ja hela poängen där är väl att man inte skall förlita sig på att en användare skall kunna motstå ett riktat angrepp mot en. Utan det man får fokusera på istället är	MFA EDU

	att användaren inte skall kunna dela med sig av någonting som ger någon typ av åtkomst. Så att även om jag vill ska jag inte kunna. Det kan ske med någon form av hård token tex för inloggning. Man skall inte kunna lämna ut ett lösenord för man ska inte själv veta vad det är för lösenord. I windows och microsoftmiljöer kan man då använda windows hello tex, den har en pinkod men utan ditt ansikte spelar det ingen roll och så vidare. Där sker också en förändring, man börjar se mer på att det här att utbilda användarna, att försöka identifiera vad är supportorganisationer, vad är någon utomstående som försöker luras? Det är inte riktigt rätt väg att gå, iallafall inte att bara förlita sig på det. Utan tekniken skall vara så pass säker att användarna inte har möjlighet att bli lurade. Det är väl en sån förändring som pågår nu iallafall. Bort med lösenord. Bort med sådana saker. Tvinga användare till tokens eller biometrilösningar.	AWA ZT FT
12	***Finns det några unika aspekter av svensk kultur eller samhälle som har påverkat hur databassäkerhet hanteras i landet?***	
12	Lite patriotisk vill man väll tro att Sverige är bättre på säkerhet än andra länder, men jag vet inte om jag har så mycket belägg för det. Samtidigt som man ser till de här stora dataläckagen som har skett så är det väldigt sällan det har varit mot svenska sidor, utan det har varit mot stora internationella - och det kan också bero på att de är såklart smaskigare mål då än de mindre svenska - men aa, jag vet inte, kan det vara en skillnad att vi gärna sätter upp våra egna system medans andra då lägger det hos amazon och sånt där i större utsträckning. Det blir mer självhostat här så att man har inte sin EC2 tjänst eller sin bucket som man häller in all data i och så ligger den öppen för allmänheten att tråla i. Jag har inte stött på så många svenska organisationer iallafall som har blivit ertappade med att ha data i publika buckets då. Det är lite för dåligt underlag för att kunna säga någonting, det är mer en känsla bara.	CUL
13	*Frågade inte*	
14	***Vilka nya teknologier eller trender tror du kan ha den största inverkan på databassäkerheten under de kommande åren?***	
14	Just som jag var lite inne på, det här med att nu börjar tvåfaktorslösningen bli lättare. När du köper en dator idag så har den ofta någon form av biometri, det kan vara en IR-kamera som mappar ansiktet, det kan vara ett fingeravtryck, det kan vara ett smart card slot som nu börjar, precis när det börjar få ett fotfäste börjar det försvinna från våra datorer för de är för små för att ha ett stort smart card i sig. Men den typen av lösningar för åtkomst till system kommer ju öka upp säkerheten markant mot att ha användarnamn och lösenord. Det är ju också då aa, vad man ska se på designen av det hela, det är väl också medvetenheten om att det här kan inträffa. Det finns ju produkter som tex har loggning av databasaktiviteter. Så att när man gör en databasdump till exempel så kan den larma för att den känner igen beteendet av en händelse. Tidigare har det varit systemkonto som gjort backup av databasen, nu är det en individ eller ett personligtkonto som visserligen är administratör, och larmar på grund av det för att det aldrig inträffat förut och borde inte inträffa. Det finns många sådana här lösningar som kommer, men som allt annat det tar ju resurser att - all säkerhet ju har resurser, så det blir en avvägning där, hur mycket man vill	MFA FT AM

	lägga på att härda och säkra mot att bara få saker att fungera. Men framöver skulle jag nog säga att komma ifrån det här konceptet med lösenord är en väldigt stor fördel. Om jag fick välja en åtgärd så skulle jag säga, använd inte lösenord.	
14	***Följdfråga: Nu med llm's och så som chatgpt och så vidare; ser du att det finns en kommande hotbild från det genom tex phishing och social engineering, att man kan liksom bygga ett llm som interaktivt kan fiska hos tio hundratusentals människor hela tiden, och inte bara genom chatt och mail men också genom voice och när AI blir ännu bättre video osv.***	
14	Ja absolut när det kommer till textbaserade, det finns tyvärr så otroligt många kriminella så där finns det tillräckligt mycket med människor att göra de här typer av phishing textmässigt. Så att visst en AI skulle ju kunna - det skulle vara en effektivisering, en first line att få någon på kroken, att kunna svara lite på frågor för att sen eskalera det till en riktig person för att driva det i hamn. Men just när det kommer till emulering av röst och liknande, då börjar det bli lite klurigt beroende på hur mycket publika personer med mycket data att träna på, där står man ju för en större risk. Att emulera någon vanlig persons röst är betydligt svårare, man måste ha ett större underlag. Men säg att VDn för ett stort bolag ringer upp någon och säger att jag kommer inte in, ringer till hjälpdesken. Återställ mitt lösenord, gör det här, det är klart att det är mer troligt att någon går på det då om man känner igen den här personen. Där kommer AI vara ett problem framöver, just för att det blir lättare att luras, men då faller man ju tillbaka då till att man ska inte ha ett system som tillåter en användare att bli lurad. Bättre sagt än gjort men på vissa ställen har man lyckats iallafall. Man ska inte ha någonting att bli lurad på.	FT ZT MFA
15	***Baserat på din erfarenhet, vad skulle du rekommendera som bästa praxis för organisationer i Sverige för att förbättra sin databassäkerhet framåt?***	
15	Det är svårt att sätta någon priolista men det är en sak man sparar mycket huvudbry på, om man inte behöver ta hänsyn av användarens skicklighet i att upptäcka phishing eller inte, utan att tekniken löser det åt en åt samtliga. Men annars är det ju då att inte fokusera så mycket på säkerhetsprodukter utan mer att försök härda upp en miljö, få en typ av baseline av ett normalläge, och sen försök hitta smarta datakällor där man kan upptäcka intrång och hantera det då. Anomalier vid autentisering och åtkomst till olika tjänster och liknande. Det är sånt som ger mer. Det traditionella med att ha en regex för att upptäcka en konstig query till en webbsida, det funkar inte längre utan man måste ha andra metoder. Och då är det mer att ha koll på din miljö så att du upptäcker anomalier. Och sen kan det gå hur långt och djupt som helst att försöka åstadkomma det, men det måste ändå vara en typ av mål. Och sen så finns det såklart de tråkiga och normala som hjälpt i alla tider; ge inte högre rättigheter än vad man ska göra, man ska inte vara administratör på den dator som man jobbar på utan man ska vara det på ett annat ställe osv. Det finns väldigt mycket där. Men börjar man med det, stark autentisering, härdade klienter, god detektionsförmåga på klienterna så kommer man långt.	EDU FT WA PA

16	***Finns det något annat som du vill tillägga?***	
	Om fokus är databaser så blir det det ju också att databasen i sig är ju bara en källa, den i sig ska ju inte göra någonting utan det är alla interaktioner och tjänster mot den som blir vägen in dessvärre. Man får lägga fokus på att det är dom som man får skydda. Databasen skall ju bara tillhandahållas, ställer man en fråga jag vill ha den här tabellen då ska man få den. Sen ska sakerna run-tomkring vara någon form av filter. Och med det sagt, använd inte en databas för allting som man har i en organisation, splitta gärna upp det så att intrång på en inte leder till intrång på dem andra. Det är en bra sak att tänka på	SM

Part D: Interview Anders Hjortberg – Tetra Pak

Interview participant: Anders Hjortberg
Title: System Infrastructure Specialist
Organization: Tetra Pak
Past experience: Software Engineer at connectBlue AB, Software Engineer at u-blox, IT/IS Manager with the Swedish Armed Forces
Date: April 20th, 2023
Method: In person meeting at Tetra Pak
Length: 37:16

Q	Answers & Follow up Questions	Code
1	***Namn, arbetstitel och organisation?***	
1	Anders Hjortberg, System Infrastructure Specialist på Tetra Pak Packaging AB	
2	***Hur länge har du varit verksam inom denna organisation?***	
2	7 år, om en månad ungefär.	
3	***Vad har du för ansvarsområden i dagsläget?***	
3	Vi är ett team som utvecklar konnektivitetslösningar och någonting som vi kallar Plant och Equipment Gateway som är gatewaylösningen och det är teamets ansvar för att utveckla den här teknologin, så vi har egentligen inga individuella ansvarsområden på det viset då vi jobbar agilt i ett scrum-team. Alla tar ansvar för produkten, för helhetslösningen. Test, kvalite, säkerhet och så vidare.	
4	***Hur länge har du varit verksam inom området cybersäkerhet i Sverige?***	
4	Sedan 98 skulle jag nog säga.	
4	***Följdfråga: Vad har det varit för cybersäkerhet du varit involverad i sedan innan Tetra Pak antar jag.***	
4	Ja innan Tetra Pak, mellan 98 och 2001 så var det inom svenska försvarsmakten både i sverige och utomlands och sen mellan 2001 och 2016 så var det för ett, jag jobbade på IT-avdelningen för ett litet företag som hette Connect Blue som sen blev uppköpta av några som hette Ublox och då jobbade vi med IT-säkerhet och IT generellt.	
5	***Vad var de vanligaste säkerhetshoten mot databaser när du började din karriär?***	
5	Ja det är ju normalt sätt så var det ju nätverkssäkerhet men jag skulle säga att den största problematiken förr, det var att det fanns en benägenhet att lagra användarnamn och lösenord i klartext i databasen. Vilket kvarstår än i dag på många ställen. Men det är någonting man måste jobba med såklart. Men annars är det ju fysisk säkerhet har vi ju haft rätt så bra koll på i många årtionden. Alla vet att det är viktigt att ha larm på sin företagslokal till exempel. Man ska ha en	HIS HS SM

	serverrum som har begränsad access. Men att säkerställa nätverksaccesen har ju varit ett område man har jobbat på mer och mer över en lång period. Och så fort WiFi kom in i bilden så var det många som glömde bort att WiFi var ett sätt att accessa infrastrukturen på och därav databas access men sen finns det ju såklart om man haft webbapplikationer och sånt så har dom här cross site scripting grejerna varit ett stort hot också, lite mindre idag.	
6	***Hur har landskapet för säkerhetshot mot databaser förändrats i Sverige under tiden du varit verksam?***	
6	Jag tror det är större fokus och större medvetande om säkerhetsaspekten när det kommer till att skydda sin data. Och där jag har varit inblandad så har det ju varit stort fokus på att både skydda datan i databasen och skydda den från otillbörlig access. Det är ju rätt stort idag till exempel, det finns någonting som heter zero-trust, som är ett ramverk egentligen och det växer mer och mer med tiden. Men det är ett arbetssätt, som jag själv har jobbat med sedan 2010, det är att man i grund och botten handlar det om att man stänger ner allt och sedan öppnar upp till dem som behöver access men då måste man också veta varför dem behöver access och till vad.	AWA ZT HIST
6	***Kommentar: Så att minimera risken för att, om det sker ett intrång, så kanske det ändå inte läcks något?***	
6	Ja det finns många olika sätt att göra det på. Ett sätt är att kontrollera accessbehörigheter men sen finns det också jobb som är indirekt skyddande, det är ju sådär nätverkssegmentering och användning av VLAN och så vidare.	SM ZT
7	***Vilka har varit några av de mest betydande incidenterna gällande databassäkerhet i Sverige, och vilka lärdomar drogs av dessa erfarenheter?***	
7	Hehe, jag vet inte, jag har inget bra svar på det. Men jag kan, sen var det heller inte om det var några några direkta lärdomar dragna ur det men regeringens schabbel med trafikregistret var ju en ganska stor grej, men jag tyckte ju också att den tystades ner rätt fort så jag vet inte om det blev några lärdomar av det.	HIS
7	***Följdfråga: Har du någon insikt av vad som skedde vid den incidenten? Överhuvudtaget i breda drag?***	
7	Ganska begränsat, det var egentligen inget som intresserade mig. Jag såg att ansvarig minister blev utsparkad men att han sen kom tillbaka igen så att därav att inga lärdomar drogs.	
8	***Vilka var de viktigaste tekniska framstegen eller metoderna som bidrog till utvecklingen av databassäkerhet i Sverige?***	
8	Det var en bra fråga om jag kan nämna något specifikt, det jag vet inte. Jag tror bara att folk har bara blivit mer och mer medvetna om riskerna och att man sen har jobbat iterativt har jobbat fram olika tekniker och det är många olika tekniker som kombineras, jag skulle inte säga att det finns en som är.. löser alla problem. Utan det är ju många tankesätt eller mindset så att jag tror den största förändringen är nog egentligen att inom utvecklar communityn så har man mer	AWA HIST ZT SQL

	fokus på säkerhet från början. Man tänker på det här med zero-trust till exempel, att man inte delar ut.. man accessar inte databaser med användarnamn och lösenord nuförtiden längre, oftast, utan man försöker använda någon form av tokenbaserad access och med dom token, det finns olika teknologier så klart, men då kan man ofta begränsa olika delar av databasen vad den användaren, maskinanvändaren, appanvändaren har tillgång i databasen. Sen så man ju i många år ju också flyttat ganska mycket av frågeställningen till backenden av databasen med, vad heter det, stored procedures till exempel eller vyer. Man bygger vyerna i backenden och då kan man fråga efter information genom att anropa den här vyen. Så då har man också flyttat att det inte är klienten som ställer SQL-frågan, vilket det kan vara väldigt känsligt för går rätt lätt att modifiera den frågan.	
8	***Följdfråga: Du nämnde det här med authentication tokens, hur ser du på det här med federeringslösningar som att outsourca säkerhet till tredjeparter på det sättet?***	
8	Jag tycker det är en dålig ide. Att outsourca sin säkerhet till tredje part. Om man kan behålla den internt så är det att föredra. Ju mindre som har access till din säkerhetslösning ju bättre är det, det är min vy på det.	FED
9	***Kan du identifiera några specifika regler eller policyer som har spelat en avgörande roll för att forma databassäkerheten i Sverige***	
9	Nej. Tyvärr.	
9	***Följdfråga: Tycker du det hade behövts?***	
9	Nej. Jag tycker att det är klart det ska finnas policy och guidelines men ska dom vara tvingande, nä det tycker jag inte. Sen visst kan det finnas guidelines.. Det finns.. ja okej det har ju inte direkt med databassäkerhet att göra men indirekt, det finns ju olika ramverk, vi använder ju den här, vad heter den här IEC 62443 eller vad dom nu heter, den gamla gamla ISO99 standarden som bytat namn.. jag har glömt namn på den nu. Den försöker vi implementera nu på Tetra Pak till exempel och fler och fler kunder kräver ju att man är compliant med de här ramverken men det är fortfarande inte tvingande och jag tror inte på tvingande. Jag tror mer på att varje företag, organisation ska ha rätt att välja hur dom gör sin implementering. Sen är det ju ett förtroende man har med kunderna som sen ställer kraven, jag tror på fri marknad, gör man dåligt jobb så är man inte med längre, gör man ett bra jobb så är man med.	
9	***Följdfråga: Skulle du tycka det var bra för industrin att ha ett nationellt, kanske inte tvingande policy men ett råd eller ramverk till både den offentliga och den privata sfären i sverige? Ett gemensamt: så här sköter du databassäkerhet och best-practice?***	
9	Nä, jag tycker att branchen borde reglera sig själv och det brukar vara självreglerande eftersom det är kunderna som i slutändan som kommer ställa krav. Däremot så kan jag tycka det borde finnas någon form av information från kanske, jag vet inte, myndigheter eller branschen själv för att informera om riskerna när det kommer till IT-säkerhet just för att höja kompetensnivån hos	EDU AWA

	företagschefer och ledare inom näringslivet men även inom myndigheterna. Det tycker jag absolut hade varit nyttigt men inte tvingande. Det tror jag, sen så tror jag det kommer reglera sig ganska fort själv för när väl kunderna till dom som är användare av ditt system när dom är informerade om riskerna så kommer dom att ställa krav på leverantören och då blir det självreglerande utan att någon behöver gå in och peka med hela handen. För det brukar inte bli bra i slutändan för då om man har ett regelverk på plats som är guidelinen det är egentligen som att säga att då har företagaren ryggen fri. Då kan dom alltid peka på att "men vi har följt guidelinerna" och det tror inte jag är bra utan jag tror fri marknad och där kunderna ställer krav.	
10	***Vilka är enligt dig de viktigaste faktorerna för att upprätthålla en effektiv databassäkerhet i kontext av svenska informationssystem?***	
10	Jag tror det är kunskap att man jobbar kontinuerligt med de här frågorna och inte bara med databassäkerhet utan generiskt, så kallat cyber security. Mycket hänger ju ihopa. Det räcker inte att skydda databasen för att vara säker utan det måste vara ett helhetstänk och det är allt ifrån att göra en säker implementation av koden lika mycket som att se till så du inte ger ut access till databasen till vem som helst utan att du har koll på det. Men även att du har firewall och andra regelverk runt din databas och sen också titta över vilken data du sparar och i vilken format du sparar den i databasen. Man ska alltid ha i bakhuvet att sin databas kan bli komprometterad och då ska man helst göra det så svårt som möjligt för angriparen att plocka ut känslig data. Så det är många lager man måste tänka på om man jobbar med säkerhet. Man kan inte bara peka ut databasen specifikt tycker jag, det är allt från fysisk till nätverk till informationen i databasen.	ENC HS
11	***Hur har svenska organisationer vanligtvis ställt sig till databassäkerhet? Har detta tillvägagångssätt utvecklats över tiden? Om så är fallet, hur***	
11	Ja det här är ju min personliga uppfattning så klart och det är ju att det har varit ganska låg fokus på säkerhet generellt i det offentliga, utan många gånger så har man fokuserat på att få någonting som fungerar och det är inte jätteofta som de lyckats. Ofta brottas de med att få sina produkter att fungera. Sen så ibland så, det som hände med, det var väl transportstyrelsen, där man helt plötsligt outsourcade mycket känslig data utomlands. Det är ju ett tecken på brist på kompetens hos ledande funktion, så mer utbildning kanske eller flytta ansvaret längre ner tror jag hade varit bättre. Om det har gjorts någon utveckling på området, det kan jag inte se att det gjorts, det pratas lite mer om säkerhet men jag ser inte riktigt att de stora förvaltningarna har jobbat jättemycket med det. Det har funnits problem hos polisen, till och med hos SÄPO och det ska vara de två organisationerna som har bäst koll på sin data men det har visat sig att de inte alltid har haft det. Tror det bara var en vecka sedan där de plockade in någon som hade gjort en slagning på en gammal flickvän eller pojkvän vilket säger mig att den här personen kanske inte ska ha rätt att sitta där från början. Så säkerhet har ju också, inte bara, med utvecklingen att göra utan också med personer och individer att göra. Hade den här personen fått rätt utbildning? Troligtvis inte, för då hade de inte gjort det här misstaget. Det här är ju	CUL HIS EDU FT

	ett återkommande problem som vi har sett i medierna och jag är ganska övertygad om att vi inte får se allt i svensk media heller.	
12	***Finns det några unika aspekter av svensk kultur eller samhälle som har påverkat hur databassäkerhet hanteras i landet?***	
12	Ja, säkerhet rent generellt tror jag. Jag tror svensk kultur litar vi rätt mycket på varandra, tendens till att vara lite naiva. Och inte så det här med, på engelska "social engineering", att det inte är så stort hot men enligt statistik och siffror från säkerhetsbranschen så är ju mer än 50% av alla s.k. attacker kommer internt. Jag tror det är högre än 50%, jag tror det är närmare 70% men då får vi kolla upp dem siffrorna om vi ska ha exakt. Så det tror jag rent kulturmässigt så är det nog att vi är lite för naiva och lite för snälla och vi tror inte att finns korruption i sverige utan det är någonting som bara finns utomlands i italien och så här och det tror jag är ett problem.	CUL
12	***Följdfråga: Nu nämnde du de här negativa aspekterna med svensk kultur som naivitet och så, ser du några positiva aspekter kring svenskt, kulturellt i sverige eller hur vi jobbar med att skydda data i databaser till exempel?***	
12	Det jag kan se rent kulturellt är ju samma sak egentligen. Det är ju det, kanske inte det här med naivietet, men det är ju att vi är väldigt öppna och vi är ganska inriktade på utveckling och "innovation" i sverige och det har vi varit i, egentligen alla tider, vi har haft SAAB, Volvo, Kockums och Eriksson och andra företag här. Så det är väl den positiva sidan men när det kommer till säkerhet och så zero-trust tänket så tror jag det kan också vara negativt. Man behöver båda såklart men jag tror man behöver lite mer som företagen tänker idag att man utser security champions eller rena security team på företag som är de här pessimisterna som kan hålla alla oss utvecklare som oftast är optimister lite i schack. Vi har en tendens inom utveckling att få saker att fungera men sen hur vi får saker att fungera det är inte alltid säkert att man tänker på det, att man tänker på konsekvenserna. Utan idag så blir det mer och mer populärt att använda så kallade "threat modeling" under utvecklingsskedet där man försöker titta igenom olika attack vektorer när man skriver sin kod eller när man designar sin databas. Så det blir mer och mer vanligt och sen så har man oftast de här "security champions" och security teamen som egentligen forcerar att det sker någon form av audit innan du releasar till marknaden. Så det är ett sätt att lösa det utan att alla blir pessimister tror jag.	CUL EDU
13	***Hur står sig svensk databassäkerhetspraxis jämfört med den i andra länder? Finns det några specifika styrkor eller svagheter som sticker ut?***	
13	Det är rätt stora kulturskillnader mellan sverige och tyskland eller sverige och italien, kan egentligen bara prata om de länder jag har arbetat med och haft direktkontakt med. Tyskar där är det mer, man är väldigt noga med att följa regelverket och det var lite som vi pratade om innan, finns det ett regelverk så är det något man kan luta sig på, det betyder inte alltid att det blir bättre eller bra. Italienarna, de är mer för att undvika regelverken men ändå ge sken om att följa dem. Vi i sverige vi är också rätt så duktiga på att följa regelverk, om det finns några. Men vi är mer öppna i sverige, det är inte en säkerhetsdialog här men min spontana reaktion på kulturskillnad mellan länder är att sverige är mer	CUL

	ett platt land. Jag har inget problem att, i en grupp om vi har ett möte där chefen är med så kan vi oftast prata fritt, säga våra tanka, komma med våra ideer, bra eller dåliga, men om vi har möte med egentligen alla andra utländska bolag eller grupper, om chefen är med så dör samtalet för då är det alltid chefen som pratar och det är ingen som säger emot och det tror jag har negativ inverkan på säkerhetsarbetet.	
14	***Vilka nya teknologier eller trender tror du kan ha den största inverkan på databassäkerheten under de kommande åren?***	
14	En bra fråga. Jag ska tänka. Jag tror det går mer och mer mot decentralisering i världen, om det kommer ha en positiv eller negativ inverkan på databassäkerheten, tror till en början kommer nog vara negativ eftersom vi kommer inte ha central kontroll på datan men sen så andra sidan sett så blir det mer och mer populärt med blockchain teknologin och helt plötsligt blir det mer och mer spårbart och det blir också lättare att identifiera när någonting inte har gått som vi tänkt att det skulle gå och med den teknologin kan man också spola tillbaka vissa, om man säger, transaktioner men det kan också få konsekvenser. För frågan är egentligen hur vi förebygger att det sker istället för som den nya teknolign visar på att vi kan ha en logg som visar på att det har skett. Så att jag tror att blockchain teknologin kan ha en viss inverkan på databassäkerheten i framtiden och att jag tror att decentralisering kommer vara en utmaning för databassäkerheten. Men på sikt så tror jag att vi kommer reda ut databasfrågorna.	FT
14	***Följdfråga: Nu med ChatGPT, LLMs, och liknande produkter, till exempel vi var inne på social engineering innan, hur ser du på hotet mot, specifikt då smalt databaser från exempel social engineering via röst, video, chat med AI som kan med tillräckligt mycket data kan härma en människa?***	
14	Den så kallade "deap fake" teknologin?	
14	***Följdfråga: Ja med hoten mot databaser eller data exfiltrering?***	
14	Jag tror att det där kommer bli ett väldigt stort problem. Det finns redan idag exempel på , jag tror de flesta känner till vem Joe Rogan är, det har redan gjorts en deep fake med hans samtycke som lät ganska likt han och det finns många som kan bli lurade av det som sades så att säga och det kan ju få en ganska stor spridning på sociala medier. Det har inte direkt med databassäkerhet att göra kanske men jag tror att det har att göra med, ju mer vi kommer till den här teknologin det finns redan en massa olika teknologier som redan har funnits länge som redan lurar oss. Så TVn är ett exempel, det redigeras redan idag väldigt mycket innan vi får se någonting, det är väldigt sällan vi får se live. Vi vet ju att många högt uppsatta ledare har ju folk som agerar dubbletter av dem som dyker upp på vissa möten det har ju funnits den här teknologin med masker ganska länge som är väldigt svårupptäckta idag och nästa steg är ju att du kan göra det här med deep fake. Du kan ha spelat in någon, om det är någon som har en podcast, kan du analysera röst och tonsättning hur du uttalar vissa ord, vissa känslor och då kan du lätt, eller kanske inte lätt men du kan producera typ en intervju med den här personens röst så att det nästan är igenkännbart ifall det är riktigt eller inte. Så jag tror det är nog inget fel på teknologin utan det är nog hur vi använder den och att vi är ganska naiva som	

	människor, så har vi sett det på TV så tror de flesta att det är sant utan att ifrågasätta, utan kritiskt tänkande. Jag tror det är det största problemet med den nya teknologin. Annars tror jag på att om man tar ChatGPT och andra sådana tjänster, det kan vara både till nytta till oss och inte. Nackdelen med ChatGPT eller OpenAI eller vad de nu hette är att de nu är uppköpta av Microsoft och det är ju en jätte nackdel för helt plötsligt är det inte open source längre, på riktigt, utan nu finns där ett kommersiellt intresse, eller ett kommersiellt bolag bakom som äger teknologin och då kan de välja utveckla den i en riktning som passar dem. Jag tror att om man ska ha den här teknologin ska den vara open source där vi som utvecklare kan inspektera och göra audits på den annars så tror jag det kan bli dåligt för folket.	
15	*[GLÖMT ATT FRÅGA…]* *Baserat på din erfarenhet, vad skulle du rekommendera som bästa praxis för organisationer i Sverige för att förbättra sin databassäkerhet framåt?*	
16	***Finns det något annat som du vill tillägga?***	
16	Jag tror vi har pratat om det mesta men om det är någonting som vi borde jobba mer på inom sverige eller inom företag, det är ju att höja kompetensnivån inom bolagen och inte outsourca säkerhet, det tror jag är viktigt. Det är ganska vanligt idag, många har ju outsourcat typ inloggning till Facebook, Google och andra företag. Då har man också tappat lite av kontrollen, då får man helt enkelt lita på en tredje part utan någon insikt i deras verktyg. Jag förstår varför man gör det, det är ganska billigt och det är enkelt, de har gjort det väldigt enkelt att integrera men kostnaden är ju också att du får lite lägre säkerhet.	EDU AWA FED
16	***Följdfråga: Angående tredje parts autentisering och auktorisering skulle du vilja se till exempel en hemlagad tredje parts lösning. Om vi ska flytta ifrån tredje part helt att vara bara on-prem och alla kör sin egen lösning men om det skulle finnas en svensk version, till exmepel bankID är ett sånt, en trejde parts autentiseringstjänst, skulle du vilja se något sånt istället? Eller bara egna lösningar som du helst vill se?***	
16	Nej, asså jag tror inte att vi kan utveckla från grunden själv utan någon gång måste vi ju plocka in, köpa in teknologi eller att ta in open source teknologi. Jag är ju förespråkare för att använda så mycket opensource som möjligt för då finns där alltid en möjlighet att granska det, det är inte alltid det är möjligt för ibland så dom väldigt stora kodbaserna men man vet att det finns möjligheten att granska det och ofta så finns det ju grupper som granskar åt en, så att säga. Sen får man då lita på de grupperna så att säga eller inte, det är ju alltid det, det är svårt att dra gränsen men att utveckla allting själv från början det är ju inte möjligt. Men att outsourca till tredje part externt, det är det jag tycker man ska tänka två gånger innan man gör det. Jag ser gärna att man köper in din lösning men hosta den själv och bygg upp kompetensen runt lösningen så du förstår hur det funkar, gör egna tester, både bra och dåliga use cases du ska själv försöka, innan du väljer att ta din den, att göra lyckade autentiseringsförsök och göra icke lyckade autentiseringsförsök. Se vad som händer i de olika scenarier.	

	Sen när du har utvecklat din produkt så ser jag gärna också att man tar in en tredje part, sån här så kallad pen-testare som är experter på, ska vara experter på, att försöka ta sig förbi alla dina säkerhetslösningar. Be någon annan göra en audit på dig och få någon form utav kvitto eller validering på att du gjort ett bra eller dåligt jobb. Det är lätt att missa självklara grejer under utvecklingen så det är alltid bra att låta någon tredje part validera det du gjort och det tror jag görs i för liten utsträckning och det har att göra med att det är ganska dyrt. Men jag tror på att i det långa loppet det företag är mest rädda för, det är att förlora förtroende för sitt varumärke och göra en lösning som är dålig är en stor risk för att få en dålig rykte som skadar ditt varumärke. Så då är kostnaden högre i varumärkets förtroende och det är svårt att mäta det i ett excelark så därför tror jag det är viktigt att man, speciellt när man har externa kunder, att det är viktigt att man validerar med någon annan att man är på rätt spår. Sen så vet vi ju att är det mer än tre rader kod så är det säkert någon bugg. Speciellt när du använder bibliotek från andra, det går ju inte att skriva C# kod utan att ta in tredjepartsbibliotek till exempel.	

Part E: Interview Jesper Blomström – Cparta

Interview participant: Jesper Blomström
Title: Red Team Manager
Organization: Cparta Cyber Defense
Past experience: Cyber Security Consultant at F-Secure, IT Security Specialist at SÄPO, IT Security Consultant at Secode, System Developer at FRA, Developer at Ping Pong, Administrator and teacher at Telia
Date: April 21th, 2023
Method: Teams meeting interview
Length: 33:15

Q	Answers & Follow up Questions	Code
1	***Namn, arbetstitel och organisation?***	
1	Jesper Blomström heter jag och jag är chef för ett red team på Cparta, Cparta med C som i Cyber helt enkelt.	
2	***Hur länge har du varit verksam inom denna organisation?***	
2	Snart två år.	
3	***Vad har du för ansvarsområden i dagsläget?***	
3	Jag har ansvarat för ett red team både avseende personal men också för processer och utveckling och kundkontakter.	
4	***Hur länge har du varit verksam inom området cybersäkerhet i Sverige?***	
4	Ja, det blir ju i runda slängar, erm, men säg 15, 20 år cirka.	
5	***Vad var de vanligaste säkerhetshoten mot databaser när du började din karriär?***	
5	Men det blev ju populärt med SQL-injections och de bitarna ganska snart någon gång där på, ja efter tvåtusen så börjar det väl toppa någon gång säkert en bra bit framåt, så att det är väl de bitarna som jag tror har varit ett stort hot och att man inte har riktigt ja tänkt så mycket på segmentering och auktorisation med mera vid den tidpunkten.	HIS
6	***Hur har landskapet för säkerhetshot mot databaser förändrats i Sverige under tiden du varit verksam?***	
6	Alltså de, de antagonistiska hoten har ju blivit mer inriktade med vad de är ute efter, alltså vi pratar om en motståndare i det här laget på något sätt. APT aktörerna har ju verkligen klivit ett stort steg framåt senaste, vad vet ja, 15 åren eller någonting sådant där man har statliga organisationer som står bakom mycket angrepp. Så det har ju blivit mer inriktad vad man är ute efter för som angripare. Just hot, just databaser har ju alltid varit, vad ska jag säga, varit till för att att lagra information på ett bra sätt, så det har jag alltid varit av intresse på det sättet, så jag tror inte själva hotbilden mot databaser har på något sätt	HIS

	förändrats så jättemycket. Däremot så har det ju blivit lättare, till exempel indexera exponerade tjänster ute på nätet och rent generellt. Sen är det mycket som har hänt med på, vad ska vi säga, på utvecklingssidan då med teknologier, ramverk och språk och även, vad ska jag säga, övergången mot Cloud har också varit en stor grej.	
7	***Vilka har varit några av de mest betydande incidenterna gällande databassäkerhet i Sverige, och vilka lärdomar drogs av dessa erfarenheter?***	
7	Ja just Sverige då då? Alltså då har man ju, pratar man databaser, man kan ju säga att Active Directory är en form av databas också? Men om vi inte tänker så mycket på just den delen utan om vi tänker på det man oftast förknippar med någon form av, ja databasdumpar, som vi har pratat om någon gång tidigare och de delarna så är det ju en del stora hack, som även har påverkat Sverige, alltså om man går tillbaka till ja, tio, elva år, till LinkedIn hacket till exempel som ägde rum, så var vi väldigt många svenskar som var med i det hacket också. Så den har ju påverkat väldigt många, även Yahoo, samma sak där och så där. Sen har vi alltså hack som är mer specifikt svenska. Om jag ska försöka dra mig till minnes så var det ju till exempel Biljett.nu som var en grej en gång i tiden. Den var ganska allvarligt just för att, vilja minnas om inte jag minns fel nu, men då var ju lösenorden i klartext där. I alla fall, jag tror jag det var så. Och sen så har det även svenska påverkats av My Fitness Pal till exempel och det där delarna som ägde rum. Sen har det varit hack som inte folk känner till förstås också. Företag väljer ju att lägga locket på många gånger när de utsätts för det här, för att det är dålig publicitet. Så är det, men jag menar folkhälsoinstitutet råkade ut för någonting i samband med efter pandeminhistorien här, men det har någon vaccinerregister och så vidare. Om det är någonting mer du vill veta eller du tänker på får du styra in mig.	HIS CUL
7	***Följdfråga: Ja alltså, frågorna generellt handlar väl till exempel om, databaser är ju ett brett begrepp som du sa, ska man inkludera Active Directory eller inte. Men om vi tittar på mer traditionella databaser som till exempel SQL och NoSQL men också till exempel stora data stores eller Blobs för data. Nu nämde du några klassiska SQL databaser tror jag. Men tror du att, du som har kanske sett hos företag och myndigheter, förstår att du kanske är bunden till NDAer, men har du sett att vi har lärt oss någonting av de hacken, även de vi kanske inte kan prata om specifikt, tror du vi har lärt oss något kollektivt av de hacken?***	
7	Ja men det tror jag, jag tror verkligen det. Det är väl oftast så i livet när man går igenom någonting som svider på något sätt att man drar lärdomar av det. Vad man nu ska liksom göra för att inte råka ut för samma sak igen. Så det tror jag nog att man har lärt sig av det hela och det visar ju också lite grann på teknikutvecklingen tänker jag på när man börjar ta mer ansvar som ja, i utvecklandet av språk och utvecklande av framework så kan det ju vara så att man inte låter utvecklarna begå de här misstagen som man gjorde förr utan man styr in dem att nu kommer ni behöva använda parametriserade frågor i det här läget, ni har inget val. Så man har liksom tagit ett ansvar på den delen också.	AWA

8	***Vilka var de viktigaste tekniska framstegen eller metoderna som bidrog till utvecklingen av databassäkerhet i Sverige?***	
8	Ja, tekniska framsteg då tycker jag väl just att man automatiserar mer nu för tiden med tester och de delarna och man utvecklar då just kodnings hjälpmedel på olika sätt språken också då som vi pratade om. Så det tror jag är väldigt stora drag att det har hjälpt, bidragit till databas, ja, till att säkerheten har förbättrats man ska inte glömma medvetenheten här tror jag är viktig och den är ju inte teknisk men att man får en medvetenhet och det är ju precis som du säger att man råkar ut för någonting också så drar man slutsatser.	AWA
8	***Följdfråga: Kan du se någon generell förändring i till exempel säkrar databaser post-exploatering, till exempel hashning och saltning som vi pratat om? Har du sett någonting i det som förändrats på en kulturell eller kollektiv nivå i Sverige?***	
8	Jag tror att man har gått mot en medvetenhet där också att man får rekommendationer från, ja, det kan ju vara olika guidelines som olika leverantörer och så vidare presenterar att, du det här det så här ska du inte göra och så här ska du göra du ska ha ett salt som är unikt både för databasen som du inte förvara där du ska ha salt som är unikt för varje användare med mera. Nu var det väldigt länge sedan jag tittade på det här då, dom här delarna, så vad som händer precis idag och det senaste året, åren vet jag inte riktigt om det är så men jag skulle gissa på att det har blivit vanligare med just till exempel saltning på ett korrekt sätt eftersom man tvingas in i den fållan.	AWA HS
9	***Kan du identifiera några specifika regler eller policyer som har spelat en avgörande roll för att forma databassäkerheten i Sverige?***	
9	Nej, det är väldigt svårt. Jag har tänkt lite grann på det där och jag vågar inte riktigt säga det. Det är klart att det är i vissa fall så har man haft complience krav då som där det märker jag mer ute hos kunder nu som kontaktar oss med mera och som jag varit i kontakt med. Där har man ett compliance krav, det vill säga efterlevnadskrav att man ska ha en tredjepart som granskar den har applikationen eller nya systemet eller vad det nu kan vara. Så det är ju en form av, vad ska jag säga, policys eller regulations som man har.	
9	***Följdfråga: De här compliance kraven, ser du att det finns en stor spridning på dem? Är det unika compliance krav de har eller följer de standardiserade ramverk?***	
9	Tyvärr blir det ju väldigt mycket att företagen själva vill bara ha den där lilla bocken i kanten att de är klara med det här, att de tredjepart har granskat deras system. Så själva kraven i sig tror jag kanske inte säger så mycket, det är mer av betydelse vem som utför granskningen och vilken kompetens det finns där. Så det tycker jag är en bra grej att använda olika företag för olika, ja, köra företag A i en granskning och sedan nästa gång testar man företag B och känner man att det här är inte alls samma kvalite så då går man tillbaka till A eller så testar man företag C nästa gång för någon granskning. Sen har vi säkerhetslagstiftningen tänkte jag också lite grann på, där har ju man försökt anpassa lite grann cyberlagstifntingen till verkligheten ser ut nu med större mandat till de	

	som ska säga hanterar olika sektorer i samhället som som då på något sätt skyddsvärda. Men huruvida det här har haft någon praktisk betydelse egentligen för säkerheten i system och kanske då specifikt databas,det tror jag inte. Men ja, det är ändå ett försök till regleringar och så.	
9	***Följdfråga: Skulle du vilja se det? Tycker du att det behövs någon form utav, om vi tittar på säkerhetslagstiftningen, skulle du vilja se rena policies, regelverk eller råd som är typ på en nationell nivå till både kanske den privata och offentliga sfären om databassäkerhet, specifikt nu, men kanske cybersäkerhet eller säkerhet i allmänhet?***	
9	Ja du, nej, det korta svaret är att jag tror att det blir ett slag i luften lite grann. Man gör försök från olika myndigheters håll och organisationers håll att publicera olika best practice och do´s and don'ts på olika sätt. Det är väl en grej man har försökt att göra och jag vet inte riktigt, jag tycker det är bra och man får fortsätta att jobba på att medvetandegöra saker och ting men får parallellt med det här försöker man någon from av sanktions spår då att de företag som inte beter sig bra och slarva med saker och ting att de riskerar, ja, sanktioner, böter för att ha gjort olika saker och ting då och det kanske kan vara mer effektivt möjligtvis att de tänker att "gud alltså vi har inte råd med att riskera någon from av sanktioner, det är klart att vi måste se till att att det här har en bra, ja att våra system hanterar informationen på ett bra sätt.". Så det är svårt det där, men medvetandegörande, det är bra, man ska inte sluta med tips och råd, det är inte det jag säger. Det har alltid haltat i Sverige med samordningen runt, vad ska jag säga, nationell IT-säkerhet och det gör det fortfarande och det är lite därför vårt företag bland annat existerar. Om man tittar och jämför med andra länder där har man ett bättre samarbete mellan myndigheter än vad man har i sverige och även om man har gjort försök i sverige genom att nu ska vi ha en samordningsfunktion mellan olika myndigheter, så är det inte riktigt lyckat.	EDU AWA CUL
9	***Följdfråga: I USA har de federala organ såsom NIST, en standarsorganisation som ger kanske råd och kunskap till både den offentliga och privata sektorn. Sen har de deras olika cybersäkerhetsmyndigheter som uppfyller olika funktioner. Men du nämnde nu att vi i sverige inte har haft en bra samordning även när man prövat på de hära, nu kan man diskutera hur bra samordning USA har, men NIST är ofta en organisation man kan titta på för best practices, även internationellt, men dom riktar sig till sina inhemska system och organisationer. Men för att tolka dig rätt, du tror att även om vi försökt innan så tror du inte att det är en givande sak att ha, till exempel, en NIST för sverige.***	
9	Ja jo, men det kan ju säkert vara. Jag tror bara om man tittar hur det har sett ut senaste 15 åren så har det varit lite spretigt i Sverige, minst sagt. Man har haft med sig bl. FRA då som har haft en form av utpekat ansvar för ja, IT- och informationssäkerhet, där ska liksom expertisen inom Sverige ligga har man sagt en gång från regeringshåll. Medans man då har plötsligt en myndighet till exempel MSB som mer då har den här namnet, med samhällsskydd och den delen. Så då har man liksom den tekniska kunskapen på ett ställe, man har en myndighet som ska ha ett visst ansvar, är det tänkt, men sen har man då till exempel om man ser tillbaka så har det varit säkerhetspolisen som har haft själva	HIS CUL FT

	mandatet, om man lutar sig åt lagstiftningen, att överhuvudtaget göra dem här tillsynen att titta hur det ser ut ute hos myndigheter. Så det har haltat en del genom åren. Nu får vi se om man är på rätt väg den här gången med samordning och så vidare men jag tror att man har gjort fel lite grann nationellt från början man borde kanske ha, ja men ha pekat ut om FRA har ansvaret för IT- och informationssäkerhet nationellt ja då är det kanske de som ska hålla i själva samordningen för säkerheten inom sverige med kritisk infrastruktur och hela de bitarna. Jätte svårt område att prata om och komplext, det är inte så lätt.	
10	***Vilka är enligt dig de viktigaste faktorerna för att upprätthålla en effektiv databassäkerhet i kontext av svenska informationssystem?***	
10	Ja, du, jag tror att det där det är väl inte specifikt för svenska, utan där får man väl kanske generalisera att det är ingenting som är typiskt för Sverige men både autentisering, att man har sådana saker på plats och även auktorisation då alltså att man, vem har egentligen tillgång till den här databasen och behövs det verkligen att kreti och pleti ska komma åt alltihopa och har man en webbtjänst som snurrar som agerar mot den bakomliggande databaser, ja då ska den förstås den inte snurra med med någon form av behörigheter som skulle kunna äventyra någonting, backend. Sen är det ju, när vi har varit inne på det här som vi pratar om med saltning och även om de har lyckats med någonting som skulle kunna komma åt hela databasen ska du inte liksom ha så mycket nytta av till exempel lösenorden som är hashade då, du ska inte ha saltet som hör till databasen för det förvaras på ett annat ställe till exempel. Så, men annars är det med segmentering jag tänker på och kodgranskning. Testning är också viktigt. Nu känner jag att jag slänger ur mig en massa olika saker här men även säker utveckling, alltså kodning hur man kodar på ett bra sätt, medvetandegöra sånt. Även då fundera om man behöver kryptera information som kanske lagras i någon form av databas. Det kan också vara en grej man ska fundera på.	HS ENC EDU PA
11	***Hur har svenska organisationer vanligtvis ställt sig till databassäkerhet? Har detta tillvägagångssätt utvecklats över tiden? Om så är fallet, hur?***	
11	Svår fråga, det är väl ingen som medvetet försöker öka sårbarheter på något sätt så det är väl klart att de flesta ställer sig positivt till att det blir en ökad säkerhetsnivå, det borde jag nog påstå. Oavsett om det gäller databaser eller någonting annat då, men jag tror att kanske man har försökt de sista åren nu blivit bättre inom organisationer att fundera på, var hanterar vi information överhuvudtaget? Var har vi våra skyddsvärda system?	EDU
11	***Följdfråga: Tänker du geografiskt då eller vad är våra säkra grejer?***	
11	Ja men precis, lite säkerhetsmässigt då att man då kanske äga ett system som inte ens ska få ha någon anslutning mot någonting annat, några andra system, så att man börjar ju fundera, ok ok, vi har väldigt känslig information här och kanske inte den huvudtaget ska vara på nätet, de delarna. Att man har blivit bättre på den typen av analys tror jag har varit en utveckling över tid.	
11	***Följdfråga: Om vi tittar pre GDPR och post GDPR, ser du någon skillnad då i hur organisationer har ställt sig till databaser, för det är ju ofta databaserna som innehåller den här känsliga informationen och även om***	

	organisationen skulle bryta mot GDPRs regler mot insamlingen, om de exponeras att de har samlat in information utan samtycke när det läcks kan de riskera ganska stora viten. Har du sätt någon skillnad på post- och pre GDPR?	
11	Det tror jag absolut, att regleringen av detta, asså lagstiftningar och från olika håll och kanter har tvingat organisationer att fundera över det här med hur man hanterar de delarna. Så ja, det tror jag.	GD
12	***Finns det några unika aspekter av svensk kultur eller samhälle som har påverkat hur databassäkerhet hanteras i landet?***	
12	Jag kan inte riktigt relatera till det så, nä, jag vet inte vad jag ska svara riktigt faktiskt. Jag tror väl egentligen så här, att generellt så tror jag att vi i sverige är ganska duktiga på att strukturera och projektleda och har varit det genom åren och det tror jag vi kanske är. Jag tror vi är rätt duktiga på teknik, att vi ligger hyfsat långt framme om man jämför med en del andra länder.	CUL
12	***Följdfråga: Kan du se någon skillnad hur vi arbetar säkerhetsmässigt jämfört med andra i säkerhetsbranschen i andra länder, med liksom arbetskultur?***	
12	Oj, det var svårt. Ja, men det blir åter igen att jag får väl en känsla av det här vet inte jag om det är så att, att det känns som att Sverige har ganska lätt att testa nya teknologier och ligger hyfsat bra internationellt till vad det gäller, nä men att anpassa sig och använda de här nya språken och teknologierna som dyker upp. Man får ibland en känsla när man besöker andra webbsidor i andra länder att det kan vara att man slungas tillbaka några år i tiden beroende på vilket land man befinner sig i, så. Om man tar till exempel Sverige har legat väldigt långt fram, nu tänker jag på, med någon form av tvåfaktorsautentisering, jag tänker på bankinloggningar och de delarna. Vi har ju ganska tidigt frångått det här med skraplotter och allt vad det har varit, men det hängde nog kvar lite längre på andra ställen.	CUL MFA
13	***Hur står sig svensk databassäkerhetspraxis jämfört med den i andra länder? Finns det några specifika styrkor eller svagheter som sticker ut?***	
13	Jag kan inte svara på det måste jag erkänna. Jag vet inte.	CUL
14	***Vilka nya teknologier eller trender tror du kan ha den största inverkan på databassäkerheten under de kommande åren?***	
14	Alltså cloud migrering blir ju en sån del som kommer att påverka mycket. Man lägger sin trust, man litar på molnleverantören att den tar hand om säkerhet och alltihopa, med de delarna. Så den trenden tror jag vi kommer fortsätta se, att man frångår mer och mer, ja men från början körde de mycket on-prem när det gällde infrastruktur och sen blev det så, nu stöter man på hybridlösningarna på många ställen där man har företag som båda har en del on-prem och en del i molnet. Men nu känner vi att man stöter på ännu fler företag som går över helt och hållet till att förlita sig på en molnleverantör när det gäller användarhantering och resurser och de grejerna så den trenden tror jag att vi kommer fortsätta	FT FED

	att se. Risken där är ju att man lägger väldigt många ägg i samma korg och att man sen tillslut inte riktigt har koll på informationen och var den ligger.	
14	***Följdfråga: Det här med federering av till exempel autentiseringstjänster, vi pratade BankID och liknande saker men också de här Microsoft inloggen till allt och så vidare. Ser du det som en bra eller negativ sak att vi lägger ut det på tredje part? Vad hade du föredragit?***	
14	Allt beror på omständigheter och det är svårt att svara någonting rent generellt men magkänslan är att istället för att implementera en egen autentiseringsprocess att istället då förlita sig på en tredje part kan i många fall vara en bättre väg att gå.	FED
14	***Följdfråga: Nu med LLMer, ChatGPT och liknande ser du att det kan vara ett hot om vi pratar social engineering mot cybersäkerhet generellt, och i förlängningen databassäkerhet, att vi tar då till exempel phishing eller AI-röster eller videosamtal och så vidare. Eller massutskick av automatiserad phishing som är mer interaktiv än den varit förut. Ser du detta som ett potentiellt hot eller avfärdar du det att det inte är ett så stort hot?***	
14	Asså jag tror att vi bara har sett början på hela denna utveckling om de här delarna så att vi har nog svårt att bilda oss ett perspektiv om hur framtiden kommer att se ut för att det här kommer nog på många sätt revolutionera olika delar, både för angripare och för försvarare. Så att jag tror vi kommer att få se, ja inte bara social engineering delar men vi kommer nog att få se att man utvecklar den här teknologin för att även göra antagonistiska andra angrepp med mera. Det är nog väldigt svårt att sia om framtiden här, tyvärr, eller som tur är.	FT
15	***Baserat på din erfarenhet, vad skulle du rekommendera som bästa praxis för organisationer i Sverige för att förbättra sin databassäkerhet framåt?***	
15	Ja, men det är lite av det jag talade om tidigare med, autentiseringsdelarna och hur de är applicerade. Auktorisation, vem som gör vad egentligen och att man tittar med någon form av least privilege när det gäller databaser. Segmentering om man tittar lite mer på närverksnivå hur är allt struktuerat, sitter allt på ett platt nät eller har man segmenterat bort vissa delar. Men så då även med att hålla sig ajour med språkutveckling, ramverk, medvetandegöra saker och ting med säker kodning och så vidare. Så det är väl egentligen dom best practice, det blir många ord där då. Men det finns ingen silverkula för att lösa precis allt utan man får jobba på flera fronter parallellt.	
16	***Finns det något annat som du vill tillägga?***	
16	Nä, det är kul att se ert arbete och vart ni kommer med det här. Det ska bli kul att läsa sen. Inget annat.	

Part F: Interview Christoffer Jerkeby – Jerkeby Security Consulting

Interview participant: Christoffer Jerkeby
Title: Independent Consultant
Organization: Jerkeby Security AB
Past experience: Principal Consultant at F-secure, Senior Researcher at Eriksson, Consultant at Stickybit AB, Configuration Management at Sony Mobile Communications, Configuration Management and Security at Ericsson Mobile Platform, Hardware Development Tools and Methodology at ST-Ericsson, 3G service System Administrator at Sony Ericsson
Date: April 24th, 2023
Method: Teams meeting interview
Length: 1:14:03

Q	Answers & Follow up Questions	Code
1	***Namn, arbetstitel och organisation?***	
1	Jag heter Christoffer Jerkeby och jag är CEO på Jerkeby Security Consulting. Det är ett bolag som ägnar sig åt cybersäkerhet - konsultverksamhet.	
2	***Hur länge har du varit verksam inom denna organisation?***	
2	I ett och ett halvt år	
3	***Vad har du för ansvarsområden i dagsläget?***	
3	Jag är ju ensamföretagare, och det betyder att jag är konsult. Jag sköter marknadsföring, all kundkontakt och all sälj. Men jag sköter inte ekonomin	
3	***Följdfråga: hur ser din day to day roll ut i det du gör i ditt arbete?***	
3	Jag jobbar ute hos kunden 90% av min tid kan man säga, och då brukar det bestå i att jag gör penetrationstester och ger rådgivning. Ibland arbetar jag som en CISO och ibland är jag en stödfunktion till utvecklingsorganisationer.	
4	***Hur länge har du varit verksam inom området cybersäkerhet i Sverige?***	
4	Då har jag varit aktiv i ungefär 20 år.	
5	***Vad var de vanligaste säkerhetshoten mot databaser när du började din karriär?***	
5	20 år sen var det år 2003. Då hade buggklassen SQL-injection funnits i två år ungefär. Och den buggklassen var ju det då vanligaste hotet mot applikationssäkeret som jag minns det. Det var också prevalent och vanligt med buffer overflow attacker. Det var den vanligaste mot operativsystem, speciellt mot windowsmiljöer. Aa - det var väl de två. Mot applikationer var det SQL-	HIS SQL BO

	injection och mot services så var det buffer overflows. År 2003.	
5	***Följdfråga: SQL-injections förstår jag att det var bland annat för att exfiltrera data från databaser. Overflows kunde man också då använda för att komma åt databaser där bak. Vilka av dessa två användes mest då för 20 år sedan tycker du?***	
5	Sql injection var en attackmetod som man gjorde manuellt på den tiden. Men buffer overflows gjordes i en mycket större skala för det kunde man göra automatiskt med så kallade datorvirus. Och det hade alla. Det var ju ändå så att under några perioder så påverkades nästan hela världens windows datorer av buffer overflows, och det är ju svårt att slå någonsin igen.	HIS
6	***Hur har landskapet för säkerhetshot mot databaser förändrats i Sverige under tiden du varit verksam?***	
6	Jag tycker att det har förändrats mycket tack vare intåget av tvåfaktorsautentisering senast, det har gjort att man inte letar efter lösenord som man gjorde tidigare som angripare i lika stor utsträcking utan man letar efter sessionsinformation mer. Man handlar med informationssesion mer. Det är nyligen. Innan dess var det ju mellan 2003 och 2007 infördes salts i databaser. Man slutade köra MD5 som man gjorde från början. På grund av att salts tillfördes så uppkom en typ av crackingindustri då som kulminerade 2010 när grafikkort lanserades för knäckning. Och det förändrade ju den världen lite grann. Sen har ju också databaslagren flyttats. Där det var microsoft access i millennieskiftet användes access och oracle väldigt mycket, som togs över ungefär 2003-2004 av mysql, som numera heter mariadb. Ungefär 2008-2009 så gick databasen postgresql om mariadb i användning, och man slutade använda oracle ungefär samma tid för att deras licenskostnad var väldigt hög jämfört med de billiga databaserna. Det här gäller ju för medelstora och mindre företag främst av allt, att man hade den här typen av skiftning, men det var dem som stod för den största delen av databasbreaches. Samtidigt så finns det en typ av databasanvändare som är stordatoranvändare, alltså svenska teleoperatörer och banker och den typen, och de använder ju enterprisedatabaser. Där så förändrades säkerheten lite mer långsamt, man använde db2 i stordatorer och också oracle på grund av att de hade en bättre distribueringskapacitet. Däremot så hade dem inget som helst saltförmåga, vilket gjorde att man fick breaches. Väldigt ofta lagrade man även lösenord i klartext. Det var dumt. Det finns mer kanske men, det var inte så kronologisk ordning. Bara för att summera, vi hade bytt i databaslager i några tillfällen.. Jusste! Viktigt som tusan som jag glömde att säga. Parametrisering av SQL-queries kom ju 2004 tror jag, eller någonting sånt. Man började parametrisera SQL-queries, det gick ju inte att göra innan. Man var tvungen att konkatinera input data med SQL, så alla databaser hade injections i princip. När parametriseringen kom upp i SDKn och ramverk slutade injections vara en sak, det var ju en stor förändring.	HIS MFA HS

7	***Vilka har varit några av de mest betydande incidenterna gällande databassäkerhet i Sverige, och vilka lärdomar drogs av dessa erfarenheter?***	
7	Det är svårt att prata om vilka som var.. eh såhär.. Det fanns några publika hack som skedde i början av 2000-talet. År 2004 eller 2005 var det ju en grupp som hette the conspiracy som läckte en del databaser med hashar online. Det ledde till att många personer i hackingscenen fick upp ögonen för hur man ska gå till väga för att knäcka saker. Då hade de lagt upp lösenordslistor, så kallade shadowfiler från användardatabasen på bland annat algonet som det då hette som blev till telenor eller någonting sånt. Och i samband med det hacket tror jag att bytet av algoritmer blev en aktiv diskussion i communityt. Man pratade om att man skulle sluta med svaga algoritmer som des som man hade använt innan till exempel. Och man började byta det. Men jag vet inte om det var specifikt för Sverige, sen så fanns det parallellt med det här andra svenska hackinggrupper som var sverigespecifika. De ägnade sig åt att hacka svenska säkerhetsforskare i princip. Och då läckte det ut ett par databaser, alltså nu pratar vi från 00-05, läckte det ut ett par databaser i samband med det. Som var lite så interna forum och diskussionsgrupper och sånt. De databaserna hade ju då ingen salt, för det fanns ju inte. Och det gjorde att de här organisationerna i princip släckte ner, och jag tror att resultatet av det gjorde att många organisationer kanske började tänka till om hur de skulle publicera sin användardata. Tänk på att dataskyddsförordningen och sånt är inte så aktivt använd i den här tiden, det är i princip ingen som är anmäld för att hantera data på ett otillbehörigt sätt annat än enskilda poliser kanske. Konsekvensen av det är att det viktigaste är att man ska tänka på sitt egna rykte. När väl 2010 kom vilket är kulmen av svensk underground hacking kultur, det var då den var som störst men också samma år den dog. Då läckte det ju många databaser ut samtidigt. Flashback blev ju hackar. Flera politiska partier blev hackade av någon och det läckte ut hela databasdumpar från det. Det läckte ut databasdumpar från ett dreamhackassocierat, om det var fragbite eller någonting sånt. Något datorspelforum som läckte ut. Alla dem använde ju sig av olika typer av PHP-ramverk då. Flashback använde ju v-bulletin, och det gjorde även det här spelforumet som jag glömt vad det hette nu. I samband med att de läckte ut så händer en annan sak, och det är att priset på beräkningskapacitet från grafikkort hade ju blivit väldigt lågt. Så att från och med 2007 till 2010 så domineras helt knäckarscenen av folk som knäcker lösenord med hjälp av grafikkort. Grafikkort är väldigt väldigt bra på att knäcka lösenord som är gjorda med standardiserade kryptoalgoritmer. Det är det för att grafikkorten hade från början vissa av de här algoritmerna färdigimplementerat i sig. Om man använde openCL eller openCuda för att knäcka fick man en väldigt hög effektivitet. Det gjorde att man kunde knäcka ganska svåra lösenord på den tiden. Det här är ju också under en period när WPA-knäckning kommer igång, då kunde man använda samma typ, asså man knäckte trådlösa nät och sådär. Då använde man ungefär samma infrastruktur för att göra den här typen av knäckning. Man använde grafikkort för det här mycket mer än processorer. Processorerna var dåliga på multitråd och hade för få kärnor för att vara effektiva. Sen fanns det en tredje kategori, och det var ju då FPGA-knäckarna. Det var ju ofta forskningsrelaterade människor som visste hur man gjorde hårdvaruutveckling med så kallad VHDL-kod. Med den VHDL-koden kunde man bygga en processor som optimeras i knäckning. Det här resulterade i att man gjorde en, några exempel på	HIS HS

	hur man kunde knäcka AGSM algoritmen, kanske inte så relaterat till databaser. Men man knäckte DES och blowfish väldigt snabbt. Nu har vi alltså ett landskap där standardiserade algoritmer som tex ais knäcks av grafikkort, och mindre standardiserade eller äldre lösningar som blowfish och des, de knäcks på FPGA. På den här tiden kunde man höra sig av till organisationer med en hash och få den knäckt, man betalade ett standardbelopp. Det var innan cloud hade tagit fart, men det fanns crackingföretag som man betalade kontant för att få ett visst hash som var lite svårare knäckt. Vissa grupper var lite optimerade för det här. Det hände att kriminella hack använde sig av de här tjänsterna. Det kunde man märka eftersom att när utredarna sen sökte efter den här hashen som var knäckt så dök den upp i google svar från den här organisationen.	
7	***Följdfråga: har du några specifika saker på senare år, nu pratar vi mycket tidigt så 00-talet och sånt, men om vi tittar 2010 och framåt, har vi några svenska databasincidenter, publika eller inte, som har gjort tillräckligt stora svallvågor för att folk har börjat tänka på nytt.***	
7	Det är en som jag kommer att tänka på, och det är faktiskt gunneboläckan. Anledningen till att gunneboläckan är så känslig, är för att organisationen som hackade dem väljer att läcka ut informationen istället för att bara kryptera den. De läcker alltså ut information från gunnebo publikt på internet. Gunnebo är ett företag som tillverkar skyddsrum, så de har byggt alla dem säkerhetsrummen som man använder sig av för att förvara pengar i Sverige, och fängelse. Det här är såklart väldigt känsligt. Skisser på det här och dumpen läcker ut på internet, i samband med att gunnebo vägrar att betala. Det här är en ganska så komplex databasdump som både innehåller farlig affärsinformation, men också specifik information. Jag är inte säker på vad den innehåller för jag har inte sett den här dumpen själv, men det tror jag var en ögonöppnare för många andra kanske. Eftersom att de också var ett säkerhetsföretag, och då tror jag man inte längre pratar om att de gjorde fel, utan det kan hända vem som helst var diskussionen då. Och kan det hända vem som helst då måste man vara säker på hur man hanterar databaser.	HIS AWA
7	***Följdfråga: Du nämnde att 2007 till 2010 var kulmen av svensk hackerkultur, men också att den slutade samtidigt. Hur eller varför slutade den?***	
7	Det är svårt att veta exakt vad som hände, men jag tror att det är en transition av att två typer av kategorier av människor separeras. Den svenska hackingscenen har ju bestått av om man säger folk som flyr från parkeringsböter, hembrännare och spännande tokar. Och den andra scenen har varit akademiker som råkat träffas, det är då hackingen uppstår. Den här gruppen separeras ifrån varandra, kanske delvis på grund av klass, och delvis på grund av att vi faktiskt får en marknad för att den här akademiska delen faktiskt får jobb. Iochmed att alla de här personerna som bara hade ägnat sig åt kriminell hacking innan plötsligt fick jobb, och idag har jobb som CISO och ganska fina befattningar på svenska företag, så behövde de ju inte längre befatta sig med de här hembrännarna och tokarna som ville titta på gratis kabel-tv och inte betala för P-	HIS

	böter och sånt, som den här scenen hade utgått från. Iochmed den separationen.. Det handlar också om kanske vissa nyckelpersoner försvinner från scenen och sådär, och i samband med det så uppstår två saker; det svenska hackerspacerörelsen, som var ett lagligt sätt att prata om hacking. Och de svenska IT-säkerhetskonferenserna startar båda 2007. Och det liksom vänder i princip uppfattningen om att hacking är någonting olagligt till att det är någonting man ska göra. Ordet hack börjar representera att vara smart istället för att vara elak. Det tar ju ett tag innan staten och samhället hänger med, men tre år senare i princip 2010, då är kulmen över och svensk hacking dör som företeelse. Det sker inte så många databasdumpar och hack på det sättet utan man ser mer aktörer från andra länder engagera sig och tjäna pengar på hacking. Man håller inte på med hacking utan ekonomiska incitament längre. Det finns politisk hacking absolut men det finns inte någon hacking för hackingens skull efter den punkten och framåt.	
8	***Vilka var de viktigaste tekniska framstegen eller metoderna som bidrog till utvecklingen av databassäkerhet i Sverige?***	
8	En jätteviktig sak är introduktionen av nya skriptspråk som konkurrerade ut PHP. PHP är ett skriptspråk som är lätt att utveckla i men också väldigt lätt att göra fel i. Det gjorde man också på stor skala under åren 2000-2010. Efter dem åren så tillkommer en väldigt massa personer som lär sig utveckla i nya ramverk som är lite mer strukturerade kan man säga. Vi börjar utveckla i node.js och ruby, de ramverken tillåter inte att man gör misstag på samma sätt. Man eliminerar under några år SQL-injection klassen, sen kom den tillbaks, men den var borta under en viss period för alla lärde sig hur man skulle göra med parametrisering med ramverksstödet. Man fick också ett standardsätt att hantera databashashning, som använde bland annat salts och så. Även om det blev intrång var det mycket svårare för angriparen att knäcka lösenorden. Det här sker också under den här perioden när hackingscenen dör eller parallellt med det här. Det finns några undantag kvar som kör PHP. De som kör drupal, yumla och de som kör ekosystem runt wordpress. Där läcks det ju än idag. I princip dagligen kommer det ut en läcka från någon av de här tjänsterna för de blir hackade. Alla som fortfarande använder de här ramverken råkar illa ut. Men de som börjar använda nya ramverk har inte samma utsträckning databasläckor som de tidigare. Då började man titta på liksom effekten av att faktiskt börja om från början. Det finns ett stort community i Sverige under den här tiden som faktiskt försöker laga PHP. Bland annat i Malmö hölls det träffar där PHP-utvecklare sågs och försökte stänga så många säkerhetsbuggar som möjligt för att gemensamt göra ett bättre språk. Vi hade många som ville göra någonting bra av språket. Byggde bland annat ramverket peer som är ett bibliotek för att bland annat göra hashning och sånna saker. Men det bet ändå inte riktigt. Communityt fastnade inte på alla de här funktionerna och språket förblev fullt av olika typer av svagheter skulle man kunna säga. Vi hade också sett en segregation av de som hade lärt sig utveckla med PHP som tillhör en viss generation, och de som lär sig utveckla nya skriptspråk som är en annan generation. De som är PHPare känner såklart till sina svagheter, som kanske den nya generationen inte ens behöver tänka på iochmed att ramverket tar hand om de här sårbarheterna. Men trots det så var de nya ramverken oftast starkare och	HIS SQL HS

	hade mindre användning av tex strängkokatenering, att slå ihop två strängar med varandra innan man skickar den till underliggande tolk. Det var grundproblemet för all injectionproblematik. Cross-site scripting, sql-injection och command-injection. Alla de sårbarheterna gör man sig av med om man inte tillåter användardata godtyckligt slå ihops med databasdata innan en query. Det här löste många ramverk med parametrar, det hade man tagit höjd för när man byggde nya språk och ramverk. Sen underlättar det också att språket pearl dog, som också var lite av en bandit i början av 2000-talet. Att man hade PHP i början av 2000-talet gjorde.. Det var ju ett användar-community designat för användare med låg förståelse för kod som ändå skulle kunna sätta sig in i hur saker fungerade. Ibland gav det väldigt farliga och dumma rekommendationer. Man tvingade nästan aldrig användare att uppgradera PHP-versionen, vilket gjorde att versionen PHP-5 stannade väldigt länge. Den hade i princip inte stöd till någon annan hashingalgoritm än MD5. Många databaser fortsatte att vara MD5 väldigt länge. Därför är fortfarande tex wordpress baserat på MD5. Wordpresshashning är en slags MD5 med salt tillagt, vilket gör att det fortfarande är relativt lättknäckt. Numera på alla de här onlinetjänsterna har vi enormt komplexa lösenordskrav som man egenligen inte alls nödvändigtvis behövt ha, om man från början använde PKDF2 eller en hashingalgoritm som är lite långsammare att knäcka.	
8	***Följdfråga: vad hade du tyckt var en bra hashingalgoritm för onlinetjänster att använda?***	
8	Om det är lösenordshashning vi pratar om kan PKDF2, eller någonting som kommer från NIST-ramverket vara bra, inte bara för att det är ganska säkert. Det är ganska säkert för att den är rätt så långsam. Tyvärr är den ju lätt att knäcka på grafikkort igen eftersom den använder AIS som grundalgoritm. Men den kan ställas in att göras långsammare genom att lägga till flera rundor och antal rundor, det är en bra konfigurationsingång. Anledningen till, utöver det tekniska, är att om man bygger en onlinetjäns så vill man att användaren skall använda den tjänsten. Bland annat vill man kanske att någon ska vara en betalande kund. För att någon skall vara en företagskund måste man gå igenom någonting som kallas due-dilligence. I due-dilligence steget ställer kunden en massa krav på mig som leverantör. Om de frågar vad jag använder för hashingalgoritm och jag svarar MD5 utan en salt, då blir det ingen deal. Det är ju dumt. Det är bättre att säga vi använder den standardiserade algoritmen PKDF2 som är rekommenderad av NIST. Då har man gjort någonting som följer den trygga vägen för att det är någonting rekommenderat. Det kanske inte är den bästa algoritmen men det är den som är rekommenderad.	FT HS
8	***Följdfråga: om du hade behövt välja i en enterprisemiljö då tex, då väljer du den algoritmen tex över bcrypt för att den inte är NIST-rekommenderad?***	
8	Aa.. ja, kommersiellt gör jag det. Om det är så att du är en myndighet eller bank eller så behöver du göra det. Om det är så att du är en teknikstartup som	HS

	har ett starkt teknikfokus kan det vara ok att köra bcrypt eller titta på andra alternativ. Bcrypt har också den funktionaliteten att expandera antalet rundor, det finns också jättefina ramverk som hjälper till att räkna ut hur du ska konfigurera bcrypt just idag. Det kan ju såklart vara till en stor hjälp för att reducera chansen att någon ska kunna knäcka databasens nycklar då. Jag tycker att bcrypt har en del fördelar att den är baserad på blowfish just för att det krävs en ganska ovanlig knäckarsetup för att få vara optimerad för den, vilket är ett mer kulturellt svar än ett matematiskt svar. Men det är få som bygger FPGA setuper för att knäcka saker, det krävs en väldigt dedikerad angripare för att engagera sig där. Men om det är så att du bygger open source mjukvara kan det vara så att den mjukvaran blir populär precis som wordpress, och då kommer ju chansen öka att någon bygger en bra knäckarrigg för just den hashingalgoritmen du har valt, så det är ingen garanti heller att man har valt en ovanlig algoritm. Security by obscurity funkar sällan i längden.	
9	***Kan du identifiera några specifika regler eller policyer som har spelat en avgörande roll för att forma databassäkerheten i Sverige***	
9	Aa GDPR faktiskt. Där blir det ju nödvändigt att ha hashing minst. Det är ju ett krav. Men det kan ju också vara aktuellt att ha kryptering i databasen. Att ha ett databasanslutningslager som krypterar datan åt dig, eller att du krypterar datan innan du lägger den i databasen beroende på vilken approach du vill ta. Det finns en tredje forskningsmässig metod som man forskat lite i som kallas homomorfisk krypto. Det är alltså ett krypto som optimerats för att lägga i en databas, så man kan göra en sök query på den krypterade informationen så att man kan sortera det i alfabetisk ordning utan att berätta vad de har för tecken i namnet. Det avslöjar ju någonting om vart lösenordet börjar och så där, men med homomorfisk kryptering kan man bland annat göra numerära uppräkningar och se vem har mest belopp på kontot och göra den typen av saker som kan vara optimerande och gör tex att du kan göra site separation. Ofta när man bygger större databaslager vill man separera olika databasdelar eller shards i olika siter så att man har en större tillgänglighet. Låt oss säga att när jag loggar in på facebook så liger min facebookdata i deras europeiska availabilityzon. All data ligger aldrig överallt samtidigt, sen synkroniseras den eventuellt, kallas eventuell konsistens. Det här är möjligt med hjälp av homomorfisk krypto, så att du kan ha information som är krypterad men ändå går att mergea med annan information och kan också skickas mellan kontinenter utan att bryta GDPR. För det är ju ändå ett intressant krav. Jag skulle inte säga att homomorfisk kryptering eller hashning kommer ha en avgörande roll ändå, eftersom att den är väldigt beräkningskostsam att genomföra och den kryptografiska styrkan blir såklart lidande av att man kan "gissa" saker i informationen. Det är ju inte den hela lösningen, men det kan hjälpa för till exempel pseudonymisering, som är lite av en trend just nu. Många leverantörer som gör saker som skulle kunna missbrukas för tracking är försiktiga på grund av rådande lagstiftning att orsaka sin situation där de har mer data än de behöver, och då kan man använda sig av pseudonymisering. Att man byter ut vitala informationspunkter och representerar dem med till exempel en hash eller en UID4, som är en unik identifierare som representerar det datafältet, sen sparar man all privatlivskritisk information någon annanstans. Så man inte har så mycket transferering av	GD HS

	den privata informationen, det kan vara ett sätt att riskminimera för olika typer av brott. Jag tror att anledningen till att GDPR fungerar som lag, för att teknisk lagstiftning fungerar i regel inte, men GDPR fungerade för att det finns en stor grupp människor som bryr sig om att den skall fungera. Det sker många anmälningar och man är tvungen att ta upp fall när kunden hör av sig gör att det är en ganska lyckad lagstiftning. Det har också därför hjälpt databassäkerheten. Databassäkerhet är alltid ett last resort ellerhur, om någon har tillgång till systemet kan de då i teorin hämta all data från applikationslagret okrypterat dessutom. Men om det är så att man av en anledning som angripare bara får tag på informationen när den ligger in storage eller in memory, då får man ju databasinformationen. Och där går det att minimera risk ganska mycket på grund av nya teknologier. Eller jag, 10 år gamla teknologier.	
9	***Följdfråga: Tycker du att man hade behövt mer lagstiftning?***	
9	Jag tror att om man skulle ha det skulle man också behöva teknologier som matchar, jag vet inte vad som är först av hönan och ägget här men jag kan uppleva att svenska företag har redan svårt att följa GDPR. Väldigt svårt faktiskt. Det är svårt att uppnå de kraven som finns. Det som är så fantasktiskt är att riktlinjerna är så tydliga så folk har vetat vad det är de ska göra. Hade man velat införa ytterligare direktiv, tex som NIS direktiv som NIS2 och försöka få det att följas upp, där man i NIS direktiven så pekar man ju ut då känsliga företag som är bärande för sveriges samhälle på något sätt, och ställer lite högre krav på att de ska ha en viss typ av testning och bakgrundskontroller och sådär. Hade man velat göra någonting sånt fast på ett databaslager, ni som bär på svenska databaser som tex simkortsdatabaser och den typer av saker. Ni måste följa den här typen av direktiv så att inte allt läcker ut på en gång, det hade varit en klok sak att göra. Då hade man behövt ha tydliga riktlinjer som ändå är generella. De måste ju fungera för den lilla firman såväl för som den stora firman. Det får inte vara dyrt. De måste ju också vara generella nog för att gälla alla företag, så att den inte säger du måste använda PKDF2, vilket kanske är ett konstigt krav för att det inte passar i alla fallen. Det kanske gör att vissa saker blir omöjliga, PKDF2 är långsamt och då hade det inte fungerat på IOT till exempel. Den här typen av avvägningar gör att man inte kan vara så teknologispecifik, men man kan inte vara för generell för då händer ingenting. Samtidigt så vill man ju värna om möjligheten att vara liten och stor aktör. Därför behöver man mäta på utkomsterna när man skriver lagstiftning. Man behöver alltså titta på hur man vill att det ska bli inte hur man ska göra det. Använd databasmetoder som gör att… Tillämpa det här som gör… Sen tilldelar man då viten och straff till organisationer som inte lyckas med det här på något sätt om man då tex gör det i privatlivssyfte. Det skulle vara aktuellt att hitta en sån lösning för att skydda identifieraren och sessionen. Nu när angripare bryr sig i mindre utsträckning om lösenord idag så är de mer intresserade fortfarande att få ut så kallade cookies eller JWT tokens från tjänster de angriper. Hade man hittat en lösning där man hade kunnat kryptera de här med en hash när de arbetar i arbetsminnet på en tjänst hade det varit väldigt stor nytta, för att vanligen när en angripare lyckas ta över en server så säljer de inte längre hashar utan de säljer sessionerna på tjänsten, för att låta nya inkräktare komma in på tjänsten utan att	GD EDU HS

	ha tvåfaktorsautentisering. Hade den varit krypterad på något fiffigt sätt hade det här varit till stor nytta.	
9	***Följdfråga: Det går inte just nu då?***	
9	Jag tror det finns lösningar som har gjort det här.. Man kan till exempel om man tittar på WebAuthn har ju hittat en väldigt fiffig lösning där man lagrar en typ av hash av en handskakning som man gjort tidigare och man lagrar en publik nyckel av sin motsvarande autentiserares privata nyckel som ligger på en sticka tex. Det här är en intressant teknologi som är väldigt väldigt svår att utnyttja för en angripare, för även om jag har en signatur och en publik nyckel från en angripare så kan jag ju nödvändigtvis inte reproducera någonting av det här, när jag kommer in och vill göra saker måste jag fortfarande autentisera mig och ha nånting som representerar den här publika nyckeln, tex den privata. Det är ju omöjligt om jag inte har tillgång till den andra bäraren. Men om jag skulle få önska en så skulle jag vilja få se lagstiftningskrav på hur man tillämpar och kommer igång med MFA, alltså tvåfaktorsautentisering, för det hade tagit bort en stor del av hemligheterna i databasen. Och sen göra motsvarande för sessioner såklart, att minska ner på klartexthanteringen och cookies. Även i browsers, browsers hanterar cookies i klartext. När du tittar i c; i din temp mapp så har du dina cookies där. Det är lite av ett problem för när du går till den svarta marknaden så säljs det ofta cookies stulna från någons PC. Det hade varit bra om det inte låg på PC utan på en tredjepartstjänst. Helst inte på mobilen heller utan någonting man har i nyckelringen liksom. Men jag är väldigt försiktig med att kräva lagkrav på sånt här, därför att jag upplever att varje gång lagstiftare försöker göra rätt så blir det lite fel, och det är inte så lätt.. Ni har sett cookielagstiftning och sånt som resulterar i rutor som bara gör folk ledsna.	FED MFA
10	***Vilka är enligt dig de viktigaste faktorerna för att upprätthålla en effektiv databassäkerhet i kontext av svenska informationssystem?***	
10	Kanske kan det vara kontextseparation faktiskt. Just nu börjar många använda sig av kafka som lagringsmetod, där kafka är ju en meddelandebuss där du lägger upp meddelanden som också lagras för framtiden, där man kan säga att jag kan ett serie prenumeranter som lyssnar på ett topic och i den topicen producerar jag information. En vanlig metod är att man använder det här för en typ av audit data. Jag som användare Christoffer har nu skapat det här, detta lade man tidigare i databaser. Den typen av tjänsteseparation där man kan ha olika kanaler för olika topics är ganska fiffig. Databaslager som de har sett ut historiskt har varit så att man haft en användare i databasen och den har hetat dbuser och den har haft lösenordet dbpassword. Om en angripare kommer över de här två uppgifterna, då har dem allt. Den användaren har behörighet att göra vad som helst. Det finns ett enda användarlager. Om man bygger det enligt kafkamodellen, då tillåts olika tjänster i organisationen att prenumerera på information från olika kanaler, och hämta ut information för att spela om dem. Den typen av tjänsteseparation gör att man både kan möjliggöra en typ av zero trust där alla inte måste ha tillgång till samma databaslösning. Alla delar inte på ett	ZT

	användarnamn och lösenord utan har ett unikt inloggning till kafkan, och prenumererar på ett topic och där märker man även microtjänstarkitektur. Det tror jag skulle kunna vara nyckeln till att gå ifrån användningen av databaser i sin helhet. Man kan ha lokala messagestores i typ en redis eller mongodb för hantering av cachar och den typer av saker absolut, men när det kommer till saker som man traditionellt sätt har laglat som typ meddelanden, kommentarer, inloggningsuppgifter och den typen av saker, behöver man numera tror jag andra tjänster. Så att jag tror att nyckeln till databassäkerhet är att inte ha databaser. Och att gå ifrån databaser i en hög utsträckning om det är möjligt.	
10	***Kommentar: wow det är verkligen en intressant take..***	
10	Aa.. Det är mest för att vad man har tänkt som applikationsutvecklare i skriptspråk har ju varit att vi lägger allt i databasen, det är bra för att då kan vi återställa det från backup och sånt. Ja asså det är sant, det är väldigt bra tillgänglighet, men det gör ju också att en angripare behöver ju bara stjäla databasen och versionsnumret på tjänsten du kör, så kan de återställa allting i sin egen lokala miljö. Men om de tex får tillgång till en prenumerationstjänst i kafka så kan de på sin höjd läsa från den topicen, bara få ut information från det flödet de lyckades få access på och inte allt. Det är automatiskt accesskontrollerat. Jag tycker också att backuphantering av databaser, om jag ska ge några best practises. Väldigt väldigt vanligt är det att angripare stjäl backupfiler från databaser innehållandes saker som angriparen annars kanske inte hade fått tag på. Hur man hanterar sina backuper blir därför väldigt viktigt, och då kanske man ska tänka på att tillämpa teknik som inte lagrar kopior på samma disk eller bryter accesskontrollmodellen eller så.	FT
10	***Följdfråga: vad tycker du om, det finns ju best practises runt tex backuper. Vanligtvis olika 3-2-1, 5-3-2-regler, hur man nu vill säga det. Vad tycker du om användningen ut av användingen av så kallade immutable backups, oavsett vart de ligger off-site eller på en separat disk eller så.***	
10	Vad betyder det?	
10	***Kommentar: Nu kanske det är lite så, buzzwordgrej.. Men det är ju att man lägger data någonstans så har man liksom tjänsten som skriver data till den här off-site backupen har liksom att dens permission är döda i sisåmånga dar eller veckor eller hur man nu ställer det. Så att även om angripare kommer åt tjänsten och får full behörighet så kan de inte sträcka ut handen och liksom nukea den gamla backupen, eller förändra den.***	
10	Låter ju bra, jag har inte jobbat med backup så mycket sedan cloud kom så slutade ju folk göra det. De jobbade på andra sätt istället där liksom datan ligger kanske separerad eller det finns en klontjänst som kör någonstans hela tiden,	

	sen så har man all infrastruktur som kod. Men jag tror att det säkert skulle öppna upp vissa lagkrav, tex banker och så, om man hade den här typen av tjänst. Ordet immutability ringar ju väldigt härligt i mina öron, just känslan av att integriteten är garanterad på någonting, saker är read-once eller read-only. Det är ju en väldigt trevlig tanke. En vanlig upplevelse jag haft nu när jag försökt återställa databasbackuper efter incidenter är att man kommer tillbaks och så är datan korrupt, eller den är i en session state som inte går att använda sig. För att databasen togs till exempel vid ett tillfälle när någonting annat hände i databasen, under en import eller någonting sånt där, så kan man inte använda den här eftersom den är inkonsekvent, speciellt sql databaser blir ofta så. Då har man ingen backup, och det händer väldigt ofta i ransomware.. Det skulle jag fan ha sagt förut ja, ransomware har förändrat allt!	
10	***Kommentar: Om du vill så kan vi gå tillbaka till den frågan***	
10	Det som egentligen är viktigt, det är ju att nu på grund av ransomware har man fått sätta pengar på hur mycket ens data är värd. Jag har varit med och gjort några incidenthanteringar när det har varit ransomware, och det man vill fråga då är såhär.. Kan vi veta om kundernas information är på flykt, och om den är det vad ska vi säga till kunden, vilken information är ute? Då är det supviktigt att veta vad finns i databasen, och hur är den hashad? Då kan du veta hur mycket försprång användarna har att flytta sina lösenord eller om det låg i klartext eller hur det nu låg till. Sen så är det ju, det finns två typer av ransomware kan man säga. Den ena är de som krypterar diskar, och den andra är de som hackar och läcker. Hacka och läcka började ungefär 2007, så började aktörer läcka ut information istället för att hacka. Gunnebo var ett tydligt exempel på det. När man började med den metoden och den började tillkomma då ändrades landskapet lite för då blev det plötsligt intressant med databassäkerheten igen, för om du kan gå till ransomwareaktörernas hemsida och tanka hem en dump innehållande väldigt massa riktigt smaskiga interna lösenord som användare har. Då kan det vara så att användarna återanvänder de här lösenorden till andra saker som linkedin och sånt. Och då kan andra aktörer sitta och tanka hem den här informationen, knäcka lösenordet först. Logga in på linkedin och sitta och spamma en massa andra personer att, aa, eller försöka fiska andra personer via linkedin till exempel, eller andra tjänster. Och den typen av reuse hacking har ju blivit mer förekommande iochmed att hack and leak har blivit en större attackmetod kan man väl säga. Och det har aktualiserat värdet av lösenord men även lösenord i enterpriseorganisationer. Ibland kan man ju också som enterpriseanvändare ha konton på flera tjänster, du har kanske ett salesforcekonto som har ett annat lösenord som inte är integrerat i AD:t eller så. De kontona kanske man glömmer bort att byta fast man har samma lösenord då eller kanske ett gammalt lösenord som man hade en gång när man hade samma, så tvingades man byta på ett annat ställe, så nu är det olika lösenord heheh, så har angriparen fått tag i båda och läcker ut de här på nätet så sitter det en tredjepart och hämtar ut det. Nu när jag pratar om det här, det är en annan stor skillnad som hände också kanske 2015, det var att hela den kriminella hackingvärlden separerades från	HIS

	att man hade en aktör gjorde allt arbete till att man hade initial access brokers, och initiall access brokers är organisationer av människor som är i regel kriminella då, som hackar sig in i olika tjänster och säljer databaser eller olika typer av intial access i formen av numera tokens men tidigare lösenord. Den här grejen då säljer dem det för lite grann, så kanske ransomwareaktören får tiodubbla för sitt ransomware. Men för en initial access broker säljer man på en mycket större volym, så båda har sin plats i ekosystemet och tjänar ganska mycket pengar, där den ena är ganska dålig på att hitta zero days och komma in i system, och den andra är duktig på att kräva pengar och ha liksom telefonsupport och en massa skit. Eftersom att de ekosystem har vuxit, och sedan ukrainakriget legaliserats i Ryssland, så har ju det gjort att ryska aktörer är ju helt officiella och kan numera få hjälp av staten att göra det här och har numera legitimiserat att göra det här. Det gör ju att initial access broker scenen har fått ett ganska stort uppsving, det har visat sig vara väldigt väldigt lukrativt att ägna sig åt att sälja databascredentials. Crackade, ocrackade eller hashade liksom. I ordningen det minst värdefulla är ett ocrackat lösenord, sen ett crackat lösenord och en session är mest värdefullt. Man skiter i lösenord i en hög grad liksom. Det har ju varit en stor förändring 2015 till idag kanske, senaste 8 åren.	
11	***Hur har svenska organisationer vanligtvis ställt sig till databassäkerhet? Har detta tillvägagångssätt utvecklats över tiden? Om så är fallet, hur***	
11	På 20 år så har ju aktörerna valt, eller såhär så har ju mängden kod man behöver hålla körande har blivit mycket mycket mer. Varje person som är systemadministratör eller utvecklare både skriver mer kod och har mer kod i drift. Det gör att man har blivit bättre på att homogenisera, om man är ett företag som utvecklar i python gör man bara python till exempel. Det gör också att man homogeniserat hur man använder sina ramverk, väldigt ofta kanske man är ett företag som bara jobbar med en typ av ramverk och smalnat ner lite. I början av vårat millenium var det väldigt vanligt att man hade en netbsd maskin, en freebsd maskin, en solarismaskin, två linuxmaskiner och fyra windowsservrar parallellt liksom. Det underhållet gjorde att man hade en väldigt massa spridda och olika risker. Jag misstänker att de andra intervjuerna har varit väldigt intresserade av att prata on lanman och lösenordshanteringen i början av 90-talet som kom med windowsgrejer. Det var ju bara en av grejerna man behövde hantera då. Sen hade man också kanske risker som kan ha att göra med hur man hashade lösenord i shadowfilerna i netbsd kontra hur man gjorde i solaris och så. Detta har homogeniserats väldigt mycket, active directory har gjort några minimala framsteg i hur man gjort hashingalgoritmer och sånt. Jag kan inte svara på det lika bra som de andra har gjort så det lämnar vi. Där har det skett lite småförändringar, men på ramverksnivå om du tittar på användning av stora skriptramverk för webutveckling har det här blivit mycket mycket bättre. Ett vanligt ramverk har till exempel stöd för typ en e-commerce plattform. Allting har ju stöd för oauth till exempel, som gör att du bara behöver hantera sessioner, någon annan gör inloggningen så att du inte ens har lösenorden, skitbra grej ju. Om du ändå måste ha interna och egna lösenord har de bättre hashingalgoritmer i regel än vad de hade tidigare. Det här tycker jag, eller jag upplever det som att det blir bättre ju mer gånger man låter ett skriptspråk dö och ett nytt ta fart. Att bryta kompabilitet är jobbigt för utvecklarna	HIS

	och en stor risk för den som hade dominant och stort ramver. PHP var ju det största liksom, men att bryta med gamla dogmer gjorde att man fick en mycket bättre säkerhet och dessutom en mycket högre utvecklarvelocitet i slutändan. Det enda PHP faktiskt var bra för var att det var lättläst. Det var sämre prestanda, sämre säkerhet, det var till slut svårt att underhålla för att alla skrev på alla olika sätt. Det blev svårläst för det fanns så många olika tekniker att skriva med. Att man då hade nya ramverk gjorde ju att man kunde förkasta gamla ideer och smalna av lite hur man skulle göra, man skulle göra på ett sätt. Och därmed koncentrerar man risken i en smalare fora och kan kontrollera riskerna mer. Till exempel vi hashar alltid på det här sättet. Alltid inloggning på det här sättet. Sen så slutade man sprida riskerna på hundratals olika operativsystem och ramverk som kunde ha olika sårbarheter som är exponerade. Det här var man ju helt enkelt tvungen att göra i samband med att kodmängden ökade. Förväntningen på en webbplatform blev mycket mycket högre och då fick man mer kod och samtidigt mer och mer krav på ramverk. Och det finns några undantag, som jag inte borde nämna. Det finns Svenska företag som valt att bygga en devopsmiljö där alla utvecklare får bygga som de vill, och de har de alla fått ångra bittert.	
12	***Finns det några unika aspekter av svensk kultur eller samhälle som har påverkat hur databassäkerhet hanteras i landet?***	
12	Kul fråga! Jag tror att svenska företag har nog förknippats med att ha en hög säkerhet, ibland orättfärdigt och ibland rättfärdigt. Man har velat sälja in sig och ta rygg på varumärken som IKEA, Volvo och Ericsson kanske, som ansetts vara relativt säkert. Jag kan inte prata om några egentliga kända eller stora hack av de här tre stora företagen. Det är ganska anmärkningsvärt. Det har ju kanske inspirerat många svenska ingenjörer som blivit uppfostrade i den andan att vara lite mer riskkonservativa än de är i andra länder. Vi hade ändå under samma period vi hade en hackingpeak i Sverige så hade vi också en open source hackingpeak, vi hade mycket open source utvecklare relativt och det tillkom en växande bransch som föddes kan man säga, där Spotify liksom slår igenom, många av de här andra stora svenska undren händer samtidigt, som var ändå.. svensk innovation.. Alla dem hade ändå en ganska så smal riskaptit om man jämför med de kanske lite mer rovkapitalistiska företagen i USA som har kommit och gått med vad vi kallar för hockey sticks, unicorns och bitcoinföretag egentligen. Det finns flera exempel på företag som tagit enorma risker, med vetande att de tagit enorma risker och åkt dit för det och åkt i backen så att det smäller för det. Cryptoexchanges, cheferna åker i fängelse, den typen av grejer. Det har vi inte sett så mycket av i Europa eller Norden för den delen. Allra minst i Sverige. Riskaptiten är låg, sen kanske kunskapen också är låg. Vi har också en vana att kanske inte ha så mycket säkerhetsinnovation heller, vilket jag tror man kan behöva. Om vi skulle jämföra asså de företag jag nämde tidigare, IKEA, Ericsson och Volvo, alla de grundades på 1800-talet. Och det är våra största industrier i Sverige, och SAAB också. Men alla amerikanska största företagen är alltså google, facebook och ja dem. Alla de grundades efter 2000-talet och framåt. De har tagit enorma risker men de har också skapat väldigt mycket innovation runt till exempel identifiering, vilket har en stark koppling till databassäkerhet. Om du kan identifiera en användare med väldigt hög	CUL GD

	säkerhet, så spelar det ingen roll ens om användaren har rätt lösenord, för du kan veta om det är den personen som dyker upp. Om ni skulle välja att ge mig era gmail lösenord idag, så skulle jag ändå inte kunna logga in på era gmail konton för jag har inte eran user-agent, jag har inte eran TLS-stack, jag har inte er javaskript motor, jag har inte en sökhistorik i min chrome som ni delar. All denna information saknar jag, och det har man ju fått på grund av att man har varit innovationsföretag som har varit väldigt starka under 2000-talets start, där man har fått en slags informationsdominans. Man har varit duktig på identifiering. Den här informationen samlar inte svenska företag på i regel. Vi har ingen aktör såvitt jag vet som sitter på den här typen av information så att vi skulle kunna bygga den här typen av tjänst med hjälp av det vi har samlat in. Det gör att vi inte kan motsvara den typen av säkerhetsfunktionalitet, annat än att integrera med deras amerikanska motsvarigheter, vilket vi rimligen lagligen inte kan i många fall, för det bryter ju mot GDPR då. Är ni med på vad jag är ute efter här? Det har varit smart att samla en användarinformationen som de har gjort, man brukar säga liksom att data är den nya oljan, jag har inte alltid hållit med men i det fallet är det ju ganska användbart.	
13	***Hur står sig svensk databassäkerhetspraxis jämfört med den i andra länder? Finns det några specifika styrkor eller svagheter som sticker ut?***	
13	Jag vet inte, jag har ju bara nästan erfarenhet från svenska kunder. Det finns en skillnad i infosäk krav som man ställer på amerikanska och svenska bolag. USA har ju inte GDPR i den utsträckningen för det är ju en europeisk lag. Där finns det ju skillnader, där man liksom inte respekterar användare alls. Det är en klar skillnad. Sen finns det skillnad i hur man gör standardisering för hur man kräver säkerhet. SOC2 till exempel som är ett väldigt amerikanskt sätt att standardisera säkerhet i företag, där säger man specifikt att du måste ha exakt det här, denna kryptoalgoritmen, gärna amerikansk, det ska vara på det här sättet. Vi i Europa och speciellt i Sverige använder ju ISO27000 istället som kommer ifrån ISO. Den säger inte hur du ska tillämpa kontroll, utan att du ska ha en kontroll för tex hashning, du ska inte skicka data i klartext på något sätt, det är väldigt outcome-orienterat skulle man kunna säga. Den typen av kravställning har lett till att det finns vissa skillnader. Att använda SOC2 som är väldigt teknikspecifik, har lett till ögontjäneri, snarare än att vara outcome fokuserad som faktiskt har lett till en praktisk förbättring.	CUL ISO
14	***Vilka nya teknologier eller trender tror du kan ha den största inverkan på databassäkerheten under de kommande åren?***	
14	Jag kan tänka mig att MFA kommer göra att vi bara hanterar sessioner i databaser. Och jag kan inte svara på varför men jag tror att MFA kommer bli en så dominant faktor att vi bara kommer ha MFA till slut. Och då kommer vi bara ha sessioner, och då kommer skydd av sessioner bli att viktigare fråga. Det kommer bubbla upp lite grann. Det gäller ju också sårbarhetsklassen cross-site scripting, som också hanterar sessioner, som gör att det kommer få en annan inverkan. Det kommer behöva hanteras på nytt sätt, kanske kommer vi behöva ny typ av teknologi för cookiehantering. Det är en liten gissning. Jag kan tänka	FT MFA

	mig att företag kommer behöva göra den hära separationen jag pratade om, där de slutar lagra saker i en central databas. Man vill gärna bygga data lakes, det ska man också tror jag. Men att försöka göra det på ett "data engineering"-drivet sätt, där man då har kontextsperarerad information baserat på vilken säkerhetsklass det här har, så att man inte tar till exempel privatlivsinformation och använder den som underlag så himla mycket för sitt driv så att man kanske inte tittar på saker som skulle kunna vara känsliga i detalj, eftersom att det lagstiftas ju aktivt mot det. Där man till exempel kan bli utpekad ibland, på grund av vissa drag man har som person eller så. Ålder, kön och sånt. Sen kan jag tänka mig att den biten kommer förbättras faktiskt, för jag tycker att lagstiftningen där verkar fungera, och det är väldigt lätt att upptäcka när någon bryter mot regler. Folk är ganska vaksamma, så det är väldigt positivt. Jag hade ju hoppats på att se lite innovation i de här traditionella databaslagrena, vi hade ju noSQL-eran kan man väl säga, när man slutade med sql-statements under en period. Det känns som att den inte kommer växa längre utan att den har liksom nått sin peak. Den används ju vanligtvis för att man bygger grafdatabaser, hittar relationer mellan användare och sånt. Man byggde databaser baserat på vilka kopplingar de hade, även dem hanterade lösenordet på ett separat ställe, och det ska man nog göra med en identitetsprovider, idp, som man då använder 2FA på. Så kommer kanske grafdatabaser ha en betydelse för organisationer för företag som hanterar interaktioner mellan människor, medans organisationer som hanterar meddelanden eller meddelandehantering som tex transkaktionsbaserade saker som tellcooperatörer och banker och sånt, de behöver jobba mer med kafka och meddelandehantering och den typen av busshantering och köhantering. Där tror jag att vi kommer att se syntax och inputvalidering. Idag finns det ett jättestort problem med att stora organisationer som har många indatakällor får inkonsekvent data, vilket gör att man inte kan bygga en datadriven verksamhet. Alltså att man har olika dataformat på sin information. Personnummer hanteras olika på olika ställen, emailadresser ser olika ut, saker går inte att matcha längre, därför tror jag man kommer se en ökning av ett fall där inputvalideringen sker i databaslagret, så att när du försöker att lagra data så gör man först inputvalidering för att kolla om syntaxen är rätt format, formatet stämmer, om den är semantisk relevant, om den passar in i sammanhanget, vilket man ju bara kan göra i databasen. Bara i databasen vet du om den här personen skall få finnas där, och sen gör man accesskontroll.. Eller innan, jag vet inte, man kan göra det innan också. Ska den här personen få skriva den här typen av data? Jag har börjat se fler och fler företag välja att lägga accesskontrollen nära databaslagret med hjälp av till exempel GRPC och Kafka. Det är en tendens som jag sett och det har gjort att man också lägger inputvalideringen i det lagret, vilket gör att man kan göra datan mer konsekvent. Jag tror att det är en arkitekturell grej som kommer hända. Kommer det här till postgres också? Nej det kommer det inte. Postgres kommer inte ha datatyper.	
14	**Följdfråga: tycker du federering är bra eller dåligt och vill du se mer eller mindre av det?**	
14	Federering är ju farligt därför att du har ju då två ställen som hanterar alla sessioner, alltså både din idp-tjänst och idp-tjänsten folk faktiskt autentiserar sig	FED MFA

	med. Det finns två platser där ett brott kan ske och den har en ganska hög exponering. Men vi har också passerat en puckel av att vi haft väldigt mycket sårbarheter i OAuth 2 flöden som har blivit färre, mjukvaran har blivit lite mer härdad och har liksom gått igenom sin jungfrufärd så att säga. Det får mig att tänka att det kan nu vara så att tiden av att IDP:ers risk är låg är här. Det hoppas jag, och det förutsätter ju då att folk inte börjar använda nya ramverk eller nya mjukvarer, utan de faktiskt fortsätter att använda de här gamla, att de fortsätter använda tex keycloak som är ganska populär och blivit ganska robust som IDP. Och att man då väljer stabila ramverk, att man faktiskt får en monokultur och lite av ett monopol faktiskt. Vilket kan hjälpa säkerhet faktiskt ibland. Då skapas det normer om hur det ska vara. En open-source produkt som har nått monopolställning kan vara en positiv sak. Sen så finns det ett jättestort problem med kontohantering och det är att om det är så att en tjänst tillåter mig att autentisera mig med mitt facebookkonto och jag har ett gammalt facebookkonto som jag inte använder som har blivit breachat med längesen, så kan ju det användas för att logga in och göra saker åt mig. Det är i sig självt ett problem, så jag tycker ska man använda IDP:er i enterprisefall, jag tycker det är bra i B2B när man använder tjänster som authzero och okta som har bra 2FA stöd och bra implementerade, genomtänkta tjänster, gillar jag när det är B2C? Lite mindre. Speciellt när det kommer till till exempel just facebook faktiskt, lite mer positivt inställd är jag till google och microsoft som inte har så mycket sidointressen möjligen, eller mest microsoft faktiskt. De har ingen business i reklam. Det underlättar lite. Men de har fortfarande tillräckligt information för identifieringen. Om du använder IDP:er som identifierare, och har en 2FA, då kommer de både verifiera din 2FA, men också din identitet. De kommer säga "den här anslutningen ser inte normal ut, du använder nu en ny lur för din 2FA anslutning, den har du inte använt förut", så kan den larma på det. Den kan också larma på browsern om den inte känner igen den. Microsoft har också ganska mycket information om sina användare, vilket är en positiv sak i det här fallet, det skyddar slutanvändaren. Även om det är fruktansvärt, men det är också bra.	
15	***Baserat på din erfarenhet, vad skulle du rekommendera som bästa praxis för organisationer i Sverige för att förbättra sin databassäkerhet framåt?***	
15	Jag tycker att någonting som alla måste göra just nu är att mappa sina dataflöden för privatlivskänslig information. Man måste veta var man har sin PII. Om det finns PII i din organisation då måste den markeras i databasen, så att det är tydligt var den är och vart den har varit. Annars är det väldigt väldigt svårt att ta bort det och begära ut det right? Det här föranleder att man skulle kunna pseudonymisera information som jag pratade om tidigare. Vid varje tillfälle där man hade PII, i nästa steg då börjar lägga in en liten hash då istället, som representerar den här personen. Så separerar man ut den här saken, så att det går att göra querys på, så att en organisation kan vara datadriven utan att riskera slutanvändarens information. Att det är svårare att härleda vem det är du pratar om men ändå möjlighet att göra en typ av big data query för informationen. För att i slutändan är nästan alla svenska techbolag baserade på att göra någon form av dataanalys, man gör ju någonting med informationen man får in. Det ska man fortsätta göra, men man ska göra det utan att peka ut enskilda användares identitet. Det var liksom GDPR-svaret. Sen så kan det vara så att man	FT AWA

	hittar bra fiffiga hashingalgoritmer, man kan använda sig av open tracing för att ha koll på vem som har varit var och liksom börja mappa flöden och sånna här saker, mer detaljnivå. Ha lite koll på användarresor och i detalj faktiskt logga det som sker, och göra det i en databas som kafka kan vara en bra ide, för då kan man börja upptäcka anomalier precis som de amerikanska bolagen gör, "aha den här användaren använder nu en tjänst som aldrig använts förut, och de väljer att lägga ut alla sina pengar de har och skicka iväg dem", intressant, då kanske man kan köra ett larm, man kan börja göra anomalidetektion på dataflödet, då kan man också öka säkerheten. Att faktiskt använda datan är egentligen det man gör för att göra databassäkerheten bättre, och aktivkt hålla koll på vilken data man har var och vad som sker. Slutligen, om man vill göra sig av med SQL-injections, logga alla felmeddelanden innehållande texten "syntax error", för blir det syntax error i en SQL-query, då betyder det att det är någonting fel på den queryn, och då betyder det att det har kommit in data som har orsakat det felet, och då är det en SQL-injection. Alla fall av syntax errors i en databas bör man ha en dashboard för som går upp på en ganska hög nivå så att det är tydligt sökbart hur många syntaxfel man har om man använder SQL fortfarande, eller andra typer av SQL liknande språk också för den delen, så att man har en dashboard och presentera den både för utvecklare och för produktägare och åtgärdar sånna felen tidigt, samt utreder sånna fel tidigt. Det är då man har koll på SQL-injections.	
16	***Finns det något annat som du vill tillägga?***	
16	Nej	

References

Alwan, Z., & Younis, M. (2017). Detection and Prevention of SQL Injection Attack: A Survey, *International Journal of Computer Science and Mobile Computing*. vol. 6, no 8, pp.5-17. Available at: https://ijcsmc.com/index.php/volume-6-issue-8 [Accessed: 11 May 2023]

Anderson, J.M. (2003).Why we need a new definition of information security, *Computers & Security*, vol. 22, no. 4, pp.308–313. doi: 10.1016/S0167-4048(03)00407-3 [Accessed 12 May 2023]

Aon. (2019). Global Risk Management Survey 2019, Available at: https://www.aon.com/getmedia/8d5ad510-1ae5-4d2b-a3d0-e241181da882/2019-Aon-Global-Risk-Management-Survey-Report.aspx?utm_source=Aon&utm_medium=email&utm_campaign=Milestone%20Moments&utm_content=GRMS2019 [Accessed: 12 May 2023]

Arias, D. (2018). Adding salt to hashing: A better way to store passwords, Auth0 web blog post, Available at: https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/ [Accessed 13 May 2023).

Avgerou, C. (2000). Information systems: What sort of science is it?, *Omega*, vol. 28, no. 5, pp.567–579. doi: https://doi.org/10.1016/s0305-0483(99)00072-9 [Accessed 28 April 2023]

Bankston, K., Schulman, R. & Woolery, L. (n.d.). Case study #2: Offering two-factor authentication, New America. Available at: https://www.newamerica.org/in-depth/getting-internet-companies-do-right-thing/case-study-2-offering-two-factor-authentication/ [Accessed 11 May 2023).

Bannister, F. (2002). The Dimension of Time: Historiography in Information Systems Research, *Electronic Journal of Business Research Methods*, vol. 1, no. 1, pp.1-10. Available at: https://academic-publishing.org/index.php/ejbrm/article/view/1161/1124 [Accessed 3 May 2023].

Basharat, I., Azam, F. & Wahab Muzaffar, A. (2012). Database security and encryption: A survey study, *International journal of computer applications*, vol. 47, no. 12, pp.28–34. doi: 10.5120/7242-0218 [Accessed 11 May 2023]

BBC News. (2018). Aadhaar: 'Leak' in world's biggest database worries Indians, BBC, 5 January. Available at: https://www.bbc.com/news/world-asia-india-42575443 [Accessed 11 May 2023).

Bertino, E. (2003). RBAC models — concepts and trends, *Computers & Security*, vol. 22, no. 6, pp.511–514. doi: 10.1016/s0167-4048(03)00609-6 [Accessed 4 May 2023]

Biener, C., Eling, M. & Wirfs, J.H. (2015). Insurability of Cyber Risk: An empirical analysis, *The Geneva Papers on Risk and Insurance - Issues and Practice*, vol. 40, no. 1, pp.131–158. doi: 10.1057/gpp.2014.19 [Accessed 7 May 2023]

Black, J., Cochran, M. & Highland, T. (2006). A study of the MD5 attacks: Insights and improvements. Fast Software Encryption. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 262–277. doi: 10.1007/11799313_17 [Accessed 10 May 2023]

Brink, H. I. (1993). Validity & reliability in qualitative research, *Curationis*, vol. *16*, no. 2, pp.35-38. doi: 10.4102/curationis.v16i2.1396 [Accessed 10 May 2023]

Brottsförebyggande rådet. (n,d.). Sök statistik över anmälda brott: Gör dig egen tabell över anmälda brott. In the database: Sök utifrån 4 kap. 9 c § BrB Dataintrång. Hela landet. Perioden 1980-2022. Available at: https://statistik.bra.se/solwebb/action/index [Accessed 5 May 2023]

Bruneau, G. (2001). The history & evolution of intrusion detection. SANS Institute. Available at: https://sansorg.egnyte.com/dl/TmT2wf11v7. [Accessed 10 May 2023]

Buckman, J., Hashim, M. J., Woutersen, T. & Bockstedt, J. (2019). Fool Me Twice?, Data Breach Reductions Through Stricter Sanctions. SSRN. doi: 10.2139/ssrn.3258599 [Accessed 7 May 2023]

Carretero, J. et al. (2018). Federated identity architecture of the European eID system, *IEEE access*. vol. 6, pp.75302-75326. doi: 10.1109/ACCESS.2018.2882870 [Accessed 11 May 2023]

Chagarlamudi, M., Panda, B. & Hu, Y. (2009). Insider threat in database systems: Preventing malicious users' activities in databases, *2009 Sixth International Conference on Information Technology: New Generations, Las Vegas, NV, USA*. pp.1616-1620. doi: /10.1109/ITNG.2009.67 [Accessed 26 April 2023]

Chauhan, K. K., Sanger, A. K. S. & Verma, A. (2015). Homomorphic encryption for data security in cloud computing, *2015 International Conference on Information Technology (ICIT).* doi: 10.1109/ICIT.2015.39 [Accessed 11 May 2023]

Chesney, T., Stair, R. & Reynolds, G. (2017). Principles of Business Information Systems. 3rd ed. Andover, England: Cengage Learning EMEA.

Chirgwin, R. (2017). Sweden leaked every car owners' details last year, then tried to hush it up. The Register, 23 July, Available at: https://www.theregister.com/2017/07/23/sweden_leaked_every_car_owners_details_last_year_then_tried_to_hush_it_up/ [Accessed 12 May 2023]

Chowdhury, N., Katsikas, S. & Gkioulos, V. (2022). Modelling Effective Cybersecurity Training Frameworks: A Delphi Method-Based Study, *Computers and Security*, vol. 113, p.102551. doi: 10.1016/j.cose.2021.102551

Cimpanu, C. (2020). Hacker ransoms 23k MongoDB databases and threatens to contact GDPR authorities. ZDNet. Available at: https://www.zdnet.com/article/hacker-ransoms-23k-mongodb-databases-and-threatens-to-contact-gdpr-authorities/. [Accessed 10 May 2023]

CISA. (2021). Understanding denial-of-service attacks, *Cybersecurity and Infrastructure Security Agency CISA.* Available at: https://www.cisa.gov/news-events/news/understanding-denial-service-attacks [Accessed April 12 2023]

CISA. (n,d.). Stop Ransomware, *Cybersecurity and Infrastructure Security Agency CISA*. Available at: https://www.cisa.gov/stopransomware [Accessed 6 May 2023]

Collin, M-L. (2001). Dataintrång - en granskning av den gällande lagstiftningens omfattning och begränsning, H3 thesis, Department of Law, Lund University, Available at: http://lup.lub.lu.se/student-papers/record/1556774 [Accessed 8 May 2023]

Connolly, T.M. & Begg, C.E. (2015). Database systems: a practical approach to design, implementation, and management. 6th ed. Harlow, Essex, England: Pearson Education Limited.

Coss, D., & Samonas, S. (2014). The CIA strikes back: Redefining confidentiality, integrity and availability in security. *Journal of Information System Security.* vol. 10, no. 3, Available at: https://www.proso.com/dl/Samonas.pdf [Accessed 12 May 2023]

Cremer, F. *et al.* (2022). Cyber risk and cybersecurity: A systematic review of data availability, *The Geneva Papers on Risk and Insurance - Issues and Practice*, vol. 47, no. 3, pp 698–736. Available at: 10.1057/s41288-022-00266-6 [Accessed 20 April 2023]

Cusick, J. (2018). The General Data Protection Regulation (GDPR): What Organizations Need to Know. *CT Corporation Resource Center*. Available at: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=jZQytj8AAAAJ&citation_for_view=jZQytj8AAAAJ:WF5omc3nYNoC [Accessed 8 May 2023]

Dancuk, M. (2021). Database types explained, phoenixNAP. Available at: https://phoenixnap.com/kb/database-types [Accessed 7 May 2023]

Db-engines. (n.d.), MongoDB system properties. DB-Engines. Available at: https://db-engines.com/en/system/MongoDB [Accessed 2 May 2023]

Denning, D. E. (1987). An Intrusion Detection Model. *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp.222-232. doi: 10.1109/TSE.1987.232894 [Accessed 8 May 2023]

Dhillon, G. & Torkzadeh, G. (2006).Value-focused assessment of Information System security in organizations, *Information Systems Journal*, vol. 16, no. 3, pp.293–314. doi:10.1111/j.1365-2575.2006.00219.x [Accessed 12 May 2023]

Disterer, G. (2013). ISO/IEC 27000, 27001 and 27002 for Information Security Management. *Journal of Information Security*, vol. 04, no. 02, pp.92–100. doi:https://doi.org/10.4236/jis.2013.42011 [Accessed 10 May 2023]

Dobbertin, H., Bosselaers, A., & Preneel, B. (1996). RIPEMD-160: A strengthened version of RIPEMD, in *Fast Software Encryption.* FSE 1996. Lecture Notes in Computer Science, vol. 1039. Springer Berlin Heidelberg, pp. 71–82. doi: /10.1007/3-540-60865-6_44 [Accessed 11 May 2023]

Educative. (n.d.). What is password cracking?, Educative, Available at: https://www.educative.io/answers/what-is-password-cracking [Accessed 10 may 2023]

Elmasri, R., & Navathe, S. B. (2016). Fundamentals of database systems. 7th ed. Upper Saddle River, NJ: Pearson.

Elmrabit, N., & Yang, S-H., & Yang, L. (2015). Insider threats in information security categories and approaches, *2015 21st International Conference on Automation and Computing (ICAC)*, Glasgow, UK doi: 10.1109/IConAC.2015.7313979 [Accessed 4 April 2023]

Federal Trade Commission. (2021). Equifax to pay $575 million as part of settlement with FTC, CFPB, and states related to 2017 data breach, *Federal Trade Commission*. Available at: https://www.ftc.gov/news-events/news/press-releases/2019/07/equifax-pay-575-million-part-settlement-ftc-cfpb-states-related-2017-data-breach [Accessed 3 April 2023].

FIDO Alliance. (2018). User authentication specifications overview, Available at: https://fidoalliance.org/specifications/ [Accessed May 11 2023]

Forbes Insights. (2014). The Reputational Impact of IT Risk, Forbes. *Forbes Magazine*. Available at: https://www.forbes.com/forbesinsights/ibm_reputational_IT_risk/index.html [Accessed 3 April 2023]

Foster, J. C., Osipov, V., Bhalla, N., Heinen, N., & Aitel, D. (2005). Buffer overflow attacks: Detect, exploit, prevent. Rockland, MA: Syngress publ.

Furmanyuk, A., Karpinskyy, M. & Borowik, B. (2007). Modern approaches to the database protection, *2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. pp.590-593. doi: 10.1109/IDAACS.2007.4488489 [Accessed 12 May 2023]

GDPR-Info.eu. (2023). General Data Protection Regulation, Available online: https://gdpr-info.eu/ [Accessed 5 May 2023]

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices, *Proceedings of the forty-first annual ACM symposium on Theory of computing*, vol. 9, pp.160-178. doi: /10.1145/1536414.1536440 [Accessed 13 May 2023]

Gernhardt, D., & Groš, S. (2022). Use of a non-peer reviewed sources in cyber-security scientific research, *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, 2022, pp. 1057-1062. doi: 10.23919/MIPRO55190.2022.9803478. [Accessed 10 May 2023]

Gilchrist, J. (2003). Encryption, *Encyclopedia of Information Systems*, Elsevier, pp.87–100. doi: 10.1016/B0-12-227240-4/00054-X [Accessed 4 May 2023]

Google. (2019). Online Security Survey. Available at: https://services.google.com/fh/files/blogs/google_security_infographic.pdf [Accessed 25 April 2023).

Government.se. (n.d.) The Swedish Criminal Code, Available at: https://www.government.se/contentassets/7a2dcae0787e465e9a2431554b5eab03/the-swedish-criminal-code.pdf [Accessed 8 May2023]

Gregor, S. (2006). The nature of theory in Information Systems, *MIS Quarterly*, vol. 30, no. 3, p.611. doi: 10.2307/25148742. [Accessed 28 April 2023]

Greitzer, F. L. et al. (2019). Positioning your organization to respond to insider threats, *IEEE Engineering Management Review*, vol. 47, no. 2, pp.75–83. doi: 10.1109/emr.2019.2914612. [Accessed 10 May 2023]

Grubbs, P., Ristenpart, T. & Shmatikov, V. (2017). Why your encrypted database is not secure, *Proceedings of the 16th Workshop on Hot Topics in Operating Systems*, pp.162-168 doi: 10.1145/3102980.3103007 [Accessed May 13 2023]

Guptill, A. (2016). Writing in College: From Competence to Excellence. Open SUNY Textbooks.

Hall, M. (2019). Oracle Corporation, *Encyclopedia Britannica*. Available at: https://www.britannica.com/topic/Oracle-Corporation. [Accessed 8 May 2023]

Iacob, N.M., & Moise, M.L. (2015). Centralized vs. distributed databases. Case study', *Academic Journal of Economic Studies*, vol. 1, no. 4, pp.119-130, Available at: https://www.academia.edu/84161222/Centralized_vs_Distributed_Databases_Case_Study?f_ri=1119056 [Accessed 7 May 2023].

IBM Security. (2022). Cost of a Data Breach Report 2022, Available at: https://www.ibm.com/downloads/cas/3R8N1DZJ.

IBM. (n.d.). What is database security?, *IBM*, Available at: https://www.ibm.com/topics/database-security. [Accessed 9 May 2023]

IEEE. (n.d.). What is homomorphic encryption?, Available at: https://digitalprivacy.ieee.org/publications/topics/what-is-homomorphic-encryption [Accessed 12 May 2023]

Ingole, P.K. *et al.* (2023). Database security, *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 4, pp.1568–1576. Available at: https://doi.org/10.22214/ijraset.2023.50415. [Accessed 7 May 2023]

ISO. (n.d.-a). ISO 14000 Environmental management. ISO. Available at: https://www.iso.org/iso-14001-environmental-management.html. [Accessed 10 May 2023]

ISO. (n.d.-b). ISO 9000 Family – Quality Management. ISO. Available at: https://www.iso.org/iso-9001-quality-management.html. [Accessed 10 May 2023]

ISO/IEC 27001. (2022). Information Security Management, Available online: https://www.iso.org/isoiec-27001-information-security.html [Accessed 5 May 2023]

Jacobsen, D. (2002). Vad, hur och varför? Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen. Lund: Studentlitteratur.

Jacobsen, D.I. (2017) Hur genomför man undersökningar?: Introduktion till Samhällsvetenskapliga Metoder. Lund: Studentlitteratur.

Jain, S., & Chawla, D. (2020). A Relative Study on Different Database Security Threats and their Security Techniques, *International Journal of Innovative Science and Research Technology*, vol. 5, no. 1, Available at: https://ijisrt.com/assets/upload/files/IJISRT20JAN618.pdf. [Accessed 8 May 2023]

Jin, X., Krishnan, R., & Sandhu, R. (2012). A unified attribute-based access control model covering DAC, MAC and RBAC, *26th Conference on Data and Applications Security and Privacy* (DBSec), Jul 2012, Paris, France. pp.41-55, doi: 10.1007/978-3-642-31540-4_4 [Accessed 7 May 2023]

Johns, G. (2006). The essential impact of context on organizational behavior, *Academy of Management Review*, vol. 31, no. 2, pp.386-408, Available at: http://www.jstor.org/stable/20159208 [Accessed 3 may 2023]

Kadlec, T. (2017) The MongoDB hack and the importance of secure defaults, Snyk. Available at: https://snyk.io/blog/mongodb-hack-and-secure-defaults/ [Accessed 2 May 2023]

Kajtazi, M., Cavusoglu, H., Benbasat, I., & Haftor, D. (2018). Escalation of Commitment as an Antecedent to Noncompliance with Information Security Policy, *Information and Computer Security*, vol. 26, no. 2, pp.171–193. doi: 10.1108/ICS-09-2017-0066 [Accessed 6 May 2023]

Kaliski, B (2000). PKCS #5: Password-Based Cryptography Specification, Version 2.0. doi: 10.17487/RFC2898. RFC 2898 [Accessed 9 May 2023]

Kelley, P.G. et al. (2012). Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms, *2012 IEEE Symposium on Security and Privacy*, pp.523-537, Available at: https://doi.org/10.1109/sp.2012.38. [Accessed 12 May 2023]

Kemmerer, R. A., & Vigna, G. (2002). Intrusion detection: a brief history and overview, *Computer*, vol. 35, no. 4, pp.supl27–supl30. doi: 10.1109/mc.2002.1012428. [Accessed 4 May 2023]

Khalaf, E. (2017). A Survey of Access Control and Data Encryption for Database Security. *Journal of King Abdulaziz University Engineering Sciences*, vol. 28, pp19-30, doi: 10.4197/Eng.28-1.2. [Accessed 12 May 2023]

Kindy, D. A., & Pathan, A.-S. K. (2011). A survey on SQL injection: Vulnerabilities, attacks, and prevention techniques, *2011 IEEE 15th International Symposium on Consumer Electronics* (ISCE). pp.468-471, doi: 10.1109/ISCE.2011.5973873 [Accessed 8 May 2023]

Koerner, B. I. (2016). Inside the cyberattack that shocked the US government, Wired, 23 October. Available at: https://www.wired.com/2016/10/inside-cyberattack-shocked-us-government/ [Accessed 11 May 2023]

Lesov, P. (2008). Database Security: A Historical Perspective. *ArXiv, abs/1004.4022*. Available at: https://arxiv.org/ftp/arxiv/papers/1004/1004.4022.pdf [Accessed 5 May 2023]

Liginlal, D., Sim, I. & Khansa, L. (2009). How significant is human error as a cause of privacy breaches? An empirical study and a framework for error management, *Computers & Security*, vol. 28, no. 3-4, pp.215–228. doi: https://doi.org/10.1016/j.cose.2008.11.003. [Accessed 6 May 2023]

Lindhe, J. (2017). Transportstyrelsen röjde skyddade uppgifter – två gånger om, SVT Nyheter, 18 Juli, Available at: https://www.svt.se/nyheter/inrikes/transportstyrelsen-rojde-skyddade-uppgifter-tva-ganger-om [Accessed 12 May 2023]

Lindskog, E., Huuva, L., Lehtinen, S., & Shannon, D. (2022). Data breaches reported to the police: Offence characteristics, challenges, areas for development. English summary of report *2022:8*, *Bra.se*. Available at: https://bra.se/download/18.31d9e51d18529a09626ead/1671541872863/2022_8_Data-breaches-reported-to-the-police.pdf [Accessed 8 May 2023]

Lundgren, B., & Möller, N. (2017). Defining information security, *Science and Engineering Ethics*, vol. 25, no. 2, pp.419–441. doi:10.1007/s11948-017-9992-1 [Accessed 12 May 2023]

Mansfield-Devine, S. (2017). Meeting the needs of GDPR with encryption, *Computer Fraud & Security*, vol. 2017, no. 9, pp. 16–20. doi: 10.1016/s1361-3723(17)30100-8. [Accessed 1 May 2023]

MariaDB. (n.d.). mysql_secure_installation, MariaDB, Available at: https://mariadb.com/kb/en/mysql_secure_installation/ [Accessed 28 April 2023].

Marsh, S. P. (1994). Formalising trust as a computational concept, University of Stirling. Available at: http://hdl.handle.net/1893/2010 [Accessed 13 May 2023]

Mårtensson, R. et al. (2017). Transportstyrelsens it-affär: Detta har hänt, SVT Nyheter. Available at: https://www.svt.se/nyheter/inrikes/transportstyrelsens-sakerhetsskandal-detta-har-hant [Accessed 12 May 2023]

Menezes, A. J. et al. (2018). Handbook of applied cryptography. London, England: CRC Press.

Merkle, R.C., & Hellman, M.E. (1981). On the security of multiple encryption, *Communications of the ACM*, vol. 24, no. 7, pp.465–467. doi: 10.1145/358699.358718 [Accessed 4 May 2023]

Microsoft 365. (2019). Conditional access illustration, Microsoft, Available at: https://www.microsoft.com/en/microsoft-365/blog/2019/09/18/why-banks-adopt-modern-cybersecurity-zero-trust-model/ [Accessed 4 May 2023]

Microsoft Security. (2021). Zero Trust Adoption Report, Microsoft, Available at: https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RWGWha [Accessed 11 May 2023]

Milligan, B. (2007). The man who invented the cash machine, BBC, 25 June. Available at: http://news.bbc.co.uk/2/hi/business/6230194.stm [Accessed 11 May 2023]

Mirtsch, M., & Kinne, J., & Blind, K. (2020). Exploring the Adoption of the International Information Security Management System Standard ISO/IEC 27001: A Web Mining-Based Analysis, *IEEE Transactions on Engineering Management*, vol. PP, pp.1-14, doi: 1-14. 10.1109/TEM.2020.2977815. [Accessed 10 May 2023]

Mitev, N. (2014). The Role of History in Information Systems Research: Beyond Presentism. In: de Vaujany, FX., Mitev, N., Laniray, P., Vaast, E. (eds) Materiality and Time. Technology, Work and Globalization. Palgrave Macmillan, London. doi: https://doi.org/10.1057/9781137432124_10 [Accessed 7 May 2023]

Moriarty, K; et al. (2017). RFC 8018: PKCS #5: Password-Based Cryptography Specification Version 2.1, doi: 10.17487/RFC8018 [Accessed 13 May 2023]

Mousa, A., Karabatak, M. & Mustafa, T. (2020). Database security threats and challenges, *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*. IEEE. doi: 10.1109/ISDFS49300.2020.9116436 [Accessed 9 May 2023]

MSB. (n.d.). Ledningssystem för informationssäkerhet (LIS), MSB, Available at: https://www.msb.se/sv/amnesomraden/informationssakerhet-cybersakerhet-och-sakra-kommunikationer/standardisering-inom-informationssakerhet/lis-iso-27000/. [Accessed 10 May 2023]

MSP360. (2018). GDPR and Data Storage Management, Available online: https://www.msp360.com/resources/blog/gdpr-and-data-storage/ [Accessed 5 May 2023]

Mueller, N. (n.d.). Credential stuffing, Owasp.org, Available at: https://owasp.org/www-community/attacks/Credential_stuffing [Accessed May 1 2023]

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft, *Information and organization*, vol. 17, no. 1, pp.2–26. doi: 10.1016/j.infoandorg.2006.11.001. [Accessed 28 April 2023]

Nardi, T. (2020). All your passwords are belong to FPGA, Hackaday, Available at: https://hackaday.com/2020/05/15/all-your-passwords-are-belong-to-fpga/ [Accessed 13 May 2023]

National Bureau of Standards. (1995). FIPS PUB 180-1: secure hash standard. Gaithersburg, MD: National Institute of Standards and Technology. doi: 10.6028/NIST.FIPS.180-1 [Accessed 2 May 2023]

National Cyber Security Centre. (2021). Device Security Guidance, UK government. Available at: https://www.ncsc.gov.uk/collection/device-security-guidance/infrastructure/network-architectures [Accessed May 11 2023].

NIST. (2022). Cybersecurity Framework, Available online: https://www.nist.gov/cyberframework [Accessed 5 May 2023]

NIST. (n.d.). Implementing a zero trust architecture, NIST, Available at: https://www.nccoe.nist.gov/projects/implementing-zero-trust-architecture [Accessed May 11 2023]

Noy, C. (2008). Sampling Knowledge: The Hermeneutics of Snowball Sampling in Qualitative Research. *International Journal of Social Research Methodology*, vol. 11, no. 4, pp.327–344. doi: 10.1080/13645570701401305 [Accessed 7 May 2023]

OED Online. (2023). database, n.. Oxford University Press, Available at: https://www-oed-com.ludwig.lub.lu.se/view/Entry/47411?redirectedFrom=database& [Accessed 29 April 2023]

Ometov, A. *et al.* (2018). Multi-factor authentication: A survey, *Cryptography*, vol. 2, no. 1, p.1. doi: 10.3390/cryptography2010001. [Accessed 9 May 2023]

Oracle White Paper. (2017). Oracle Database 12c Release 2 Security and Compliance: Defense-in-Depth Database Security for On-Premises and Cloud Databases, April 2017, Available at: https://www.oracle.com/technetwork/database/security/security-compliance-wp-12c-1896112.pdf [Accessed 9 May 2023]

Oracle. (2022). *What is a database?* Oracle, Available at: https://www.oracle.com/database/what-is-database/. [Accessed 25 April 2023]

Orlikowski, W. J., & Iacono, C. S. (2001). Research Commentary: Desperately Seeking the 'IT' in IT Research – A call to Theorizing the IT Artifact, *Information Systems Research*, vol. 12, no. 2, pp.121-134. Available at: http://www.jstor.org/stable/23011075 [Accessed 6 May 2023)

Owasp. (n.d.). Access Control, The Open Worldwide Application Security Project, Available at: https://owasp.org/www-community/Access_Control [Accessed 4 May 2023]

Owasp.org. (2018). Guide to cryptography, The Open Worldwide Application Security Project, Available at: https://wiki.owasp.org/index.php/Guide_to_Cryptography [Accessed May 13 2023]

Ozsu, M. T. & Valduriez, P. (2011). Principles of distributed database systems. 3rd ed. Scholars Portal.

Perrigo, B. (2018). India has been collecting eye scans and fingerprint records from every citizen. Here's what to know, Time, 28 September. Available at: https://time.com/5409604/india-aadhaar-supreme-court/ [Accessed 11 May 2023]

Pevnev, V. & Kapchynskyi, S. (2018). Database security: Threats and preventive measures, *Advanced Information Systems*, vol. 2, no. 1, pp.69–72. doi: 10.20998/2522-9052.2018.1.13. [Accessed 11 May 2023]

Pfautsch, F. *et al.* (2020). The evolution of secure hash algorithms, *Mitteilungen*. Available at: http://dl.gi.de/handle/20.500.12116/33858 [Accessed 8 May 2023]

Pirc, J. (2017). History of intrusion detection & prevention, Secureworks, 6 July. Available at: https://www.secureworks.com/blog/the-evolution-of-intrusion-detection-prevention [Accessed 12 May 2023]

Popek, G.J., & Kline, C.S. (1979). Encryption and secure computer networks, *ACM Computing Surveys*, vol. 11, no. 4, pp.331–356. doi: https://doi.org/10.1145/356789.356794. [Accessed 11 May 2023]

Preneel, B. (2010). The first 30 years of cryptographic hash functions and the NIST SHA-3 competition, *Topics in Cryptology - CT-RSA 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.1–14. doi: 10.1007/978-3-642-11925-5_1 [Accessed 8 May 2023]

Provos, N., & Mazieres, D. (1999). A future-adaptable password scheme. *USENIX Annual Technical Conference, FREENIX Track*, vol. 1999, pp. 81-91, Available at: http://www.usenix.org/events/usenix99/provos.html [Accessed 8 May 2023]

Rivest, R. (1992). The MD5 Message-Digest Algorithm, RFC Editor, Available at: https://www.rfc-editor.org/rfc/pdfrfc/rfc1321.txt.pdf [Accessed 5 May2023]

Robson, C., & McCartan, K. (2015). Real World Research. 4th ed. Nashville, TN: John Wiley & Sons.

Rose, S. et al. (2020). Zero Trust Architecture. NIST Special Publication 800-207, National Institute of Standards and Technology, Gaithersburg, MD,. doi: 10.6028/nist.sp.800-207. [Accessed 10 May 2023]

Røset, C., Warren, V., & Chiang, C.-C. (2017). Enhanced database security using homomorphic encryption, *Information Science and Applications 2017*, pp.377-387, Singapore, doi: 10.1007/978-981-10-4154-9_44 [Accessed 13 May 2023]

Rubin, H. J., & Rubin, I. S. (2005). Qualitative interviewing: The art of hearing data (2nd ed.). Thousand Oaks, CA: Sage

Ryberg, J. (2013). Så hackades Logica, Computer Sweden, 29 April. Available at: https://computersweden.idg.se/2.2683/1.505012/sa-hackades-logica [Accessed 12 May 2023]

Safa, N. S., & Maple, C. (2016). Human errors in the information security realm – and how to fix them, *Computer fraud & security*, vol. 2016, no. 9, pp.17–20. doi: 10.1016/s1361-3723(16)30073-2. [Accessed 8 May 2023]

Santos, O. (2021). Cisco CyberOps associate CBROPS 200-201: Official Cert Guide. Hoboken, NJ: Cisco Press.

Sarmah, S. S. (2019). Database security –threats & prevention, *International journal of computer trends and technology*, vol. 67, no. 5, pp.46–53. doi: 10.14445/22312803/ijctt-v67i5p108 [Accessed 8 May 2023]

Shah, A. et al. (2019). Analyzing the Impact of GDPR on Storage Systems, *11th USENIX Conference on Hot Topics in Storage and File Systems*, p.4, Available at: https://www.usenix.org/conference/hotstorage19/presentation/banakar [Accessed 8 May 2023]

Shu, C.-C., Yang, E. Y., & Arenas, A. E. (2009). Detecting conflicts in ABAC policies with rule-reduction and binary-search techniques, *2009 IEEE International Symposium on Policies for Distributed Systems and Networks*. pp.182-185, doi: 10.1109/POLICY.2009.22 [Accessed 9 May 2023]

Shulman, A. (2006) Top ten database security threats: How to mitigate the most significant database vulnerabilities, Imperva, Inc, Available at: https://schell.com/Top_Ten_Database_Threats.pdf [Accessed 9 May 2023]

Singh, P., & Kaur, K. (2015). Database security using encryption. 2015 *1st International Conference on Futuristic Trends in Computational Analysis and Knowledge Management, ABLAZE*, pp.353-358. doi: 10.1109/ABLAZE.2015.7155019. [Accessed 12 May 2023]

Singh, S,. & Rai, R, K,. (2014). A Review Report on Security Threats on Database, *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp.3215-3219, Available at: https://www.ijcsit.com/docs/Volume%205/vol5issue03/ijcsit20140503118118.pdf [Accessed 12 May 2023]

Stallings, W., & Brown, L. (2018). 1.1 Computer Security Concepts, in Computer security*:* Principles and practice. 4th edn. New York, NY: Pearson.

Stevens, M. et al. (2017). The First Collision for Full SHA-1, *Cryptology ePrint archive,* Paper 2017/190. Available at: https://eprint.iacr.org/2017/190 [Accessed 12 May 2023]

Svenska institutet för standarder, SIS. (n.d.-a). Detta är ISO 27000 för cyber- och informationssäkerhet, Available at: https://www.sis.se/iso27001/dettariso27001/ [Accessed 2 May 2023].

Svenska institutet för standarder, SIS. (n.d.-b). Detta är ISO 9001, Available at: https://www.sis.se/iso9001/dettariso9001/. [Accessed 2 May 2023]

Svenska institutet för standarder, SIS. (n.d.-c). Detta är miljöledningssystemet ISO 14001, Available at: https://www.sis.se/iso14001/dettariso14001/. [Accessed 2 May 2023]

Svenska institutet för standarder, SIS. (n.d.-d). Säkerhetsåtgärder enligt ISO 27000-serien, Available at: https://www.sis.se/iso27001/dettariso27001/sakerhetsatgarder-enligt-iso-27000/ [Accessed 2 May 2023].

Söderqvist, N. (2021). Polisen: Därför löser vi inte cyberbrotten, *Tidningen Näringslivet*. Available at: https://www.tn.se/naringsliv/8915/polisen-darfor-loser-vi-inte-cyberbrotten/ [Accessed 8 May 2023]

Thomas, K. et al. (2019). Protecting accounts from credential stuffing with password breach alerting, *USENIX Security Symposium*, pp.1556–1571, Available at: https://dl.acm.org/doi/abs/10.5555/3361338.3361446 [Accessed 25 April 2023]

Tiwari, M. (2018). Aadhaar now world's largest biometric database: 5 facts from UIDAI CEO's presentation in Supreme Court you must know, The Financial Express, 23 March. Available at: https://www.financialexpress.com/aadhaar-card/aadhaar-now-worlds-largest-biometric-database-5-facts-from-uidai-ceos-presentation-in-supreme-court-you-must-know/1108622/ [Accessed 11 May 2023]

van Beek, J., & Gevers, R. (2020). Bcrypt password cracking extremely slow? Not if you are using hundreds of FPGAs!, Cqure.nl. Available at: https://www.cqure.nl/nl/kennisplatform/bcrypt-password-cracking-extremely-slow-not-if-you-are-using-hundreds-of-fpgas [Accessed May 13 2023]

van den Nieuwenhoff, T. (2021). Fully Homomorphic Encryption: The history, Thomas van den Nieuwenhoff, 27 May. Available at: https://tvdn.me/fhe/2021-05-27-homomorphic-encryption-history/ [Accessed 12 May 2023]

Van Der Ham, J. (2021). Toward a Better Understanding of "Cybersecurity", *Digital Threats: Research and Practice*, vol. 2, vol. 3, pp.1 1–3. doi: 10.1145/3442445 [Accessed 28 April 2023]

Venkatesh, V. *et al.* (2023). Guidelines for the development of three-level models: Bridging levels of analysis and integrating contextual influences in is research, *Journal of the Association for Information Systems*, vol. 24, no. 1, pp.65–106. doi: 10.17705/1jais.00778. [Accessed 9 May 2023]

Vergadia, P. (2022). Zero Trust and BeyondCorp Google cloud, Google Cloud Blog. Google Cloud, 10 August. Available at: https://cloud.google.com/blog/topics/developers-practitioners/zero-trust-and-beyondcorp-google-cloud [Accessed 11 May 2023]

von Solms, R. & van Niekerk, J. (2013). From information security to cyber security, *Computers & Security*, no. 38, pp.97–102. doi: 10.1016/j.cose.2013.04.004 [Accessed 8 May 2023]

Ward, R., & Beyer, B. (2014). A new approach to enterprise security, *;login*, vol. 39, no. 6, pp.6-11, Available at: https://research.google/pubs/pub43231/ [Accessed 11 May 2023]

Wetzels, J. (2016). Open sesame: The Password Hashing Competition and Argon2. In *arXiv [cs.CR]*. doi: 10.48550/arXiv.1602.03097 [Accessed 10 May 2023]

Wolford, B. (2018). What are the GDPR fines?, *GDPR.eu*. Available at: https://gdpr.eu/fines/ [Accessed 9 May 2023]

Young, A., & Yung, M. (1996). Cryptovirology: exortion-based security threats and countermeasures, *Proceedings 1996 IEEE Symposium on Security and Privacy*, pp.129-140. doi: 10.1109/SECPRI.1996.502676 [Accessed 6 May 2023]