University Of Souteast Norway

The weakest link in every system. You!

An exploration of the human problem within the IT-sector and how it affects security

Author:
Andri Fannar Arnarson

May 24, 2020

Contents

1 Introduction

1.1 Background

The early 2000s was a wild time for technology. Big changes, moving fast through the industry. Both consumer and enterprise technology changing so fast that people had to try their best just to keep up. Blink and you will miss some essential tech that will define the next short while in the industry. Moore's law keeps on being true and with faster tech comes new things to learn, new use-cases and new vulnerabilities. This rat-like race between technology and man for control has left us in a peculiar situation.

The world has never been this connected. After the rise of the internet, anyone can connect with almost anyone with the click of a mouse or the touch of a screen. We have created immensely complex systems with even more complicated hardware. Despite their sophistication, familiarity and knowledge of these systems with the general public is very low. Do not get me wrong. The amount of knowledge about these systems has never been higher. However that knowledge is spread very thin between a few individuals. We have specialized ourselves. A few people know these systems intimately. Ask Joe Blow out on the streets what capacitive touch is and he wouldn't have a clue despite probably holding the newest smartphone in his hands. But that is not Joe's fault. You can not expect everyone to know the complicated ins and outs of modern technology. Right?

Well that has been the stance of many people for a long time. Though the conclusion of this paper will most likely not be that everyone should have intimate knowledge of every piece of technology that they interact with on a daily basis, I wonder if the fact that people don't really understand the devices and technology that they use on a daily basis has an effect on how we create. Do we create systems with no regard to how people, the users, will in the end use it, their knowledge about it and why they should even be using it. Do we create tech in a way that makes it, and therefore us, vulnerable. Has the term "we can't expect people to know all this" set us up for failure?

Growing up in these times made me very aware of the technological progress we were experiencing and the rapid digitalization of our lives. You start to become very aware of how much of our lives we are giving up freely and of our own volition to companies. Companies that rely on systems that are made by people. However, despite the collective hours these people put into their work, the next big security breach is just around the corner. How can this keep happening?

[5] How can time and time again peoples data be out on the black market, yet another system "breached" or [19] storing data in plain text on an open server. How? Why? The answer to these questions is often; people. People deciding that "this was good enough" or simply not having the knowledge to do it better. The modern infrastructure of an ecommerce website, or a government organisation is more complex than what they used to land on the moon. Although there might not be as much at stake

as in 1969, a popular ecommerce website can have ludicrous amounts of sensitive user data that can be very harmful if it got into the wrong hands. And although there are laws prohibiting the mistreatment of user data, breaches are all too common. Have we built systems that do not take us into account? Can we keep on going without changing something, either in how we create systems or how we expect people to use and interact with them? How can complex systems have huge flaws in the form of one person's decision to do something. How can one person's decision compromise the data of thousands or even millions.

I would like to explore how the systems we have created all have the same inherent flaw; us! Complex systems, no matter how sophisticated, have and always will have one crutch. The people that made it and the users who use it. How much of an impact does the 'human problem' have on modern systems and how we make them? What can go wrong if we do not take the human problem into consideration and what can we do to design and create systems that will take people and how they interact with the system into account?

How can we get to the bottom of these questions? I believe in learning from others mistakes. I will choose examples of breaches or security failures within systems from the last 3 - 5 years. These examples will hopefully represent a typical system of IT-infrastructure here in Norway. I would furthermore like to use these cases to draw conclusions as to what happened, why it happened, what the consequences were and to what degree was the human problem involved. Discerning this information will hopefully give me the insight to discuss changes that I think should be made to hinder such an incident from happening again and since the human problem is inherent in everything we make, what changes should be made to lessen its effects. These changes will be based off of standardized procedures recommended by select and well appraised IT-security firms and groups.

1.2 Scope

The scope of such a topic is quite important. Setting it too wide will render it ineffective for one person to do. And setting it too narrow will risk it not being representative. As I have previously stated, I will be choosing examples of breaches or "incidents" in Norway, from the last 3 - 5 years, where user data has been at risk, misplaced, compromised or erased. These incidents should be representative of modern systems that either users or developers interact with quite often as to be relevant for as many people as possible. I will be limiting myself to one case, so as to not exceed my limitations. This means that the scope is quite narrow. I will have to take good care to, again, choose an example that can represent modern IT-systems that are either a common use case or very common for a typical Norwegian person to use. Something that many people interact with on a daily basis.

The reasoning for these requirements is threefold. Firstly: I feel that examples that represent either systems that are widely used or impact the most people if compromised will resonate better the severity of the situation and what impact it can have

on people's lives. If you have used or still use some product or service that has been breached you are more likely to reevaluate either how you use said product or service or even whether to use it at all [14]. Secondly: Since I am only choosing one case, and I want to maximise the effectiveness of that case as an example, I will have to take great care in my choice.

And Thirdly: Every incident is as unique as every system is. So as I investigate the case, I will take note of how, why, when and where things happened and evaluate if it is at all relevant to this paper. I will try to contact someone that has relevant experience connected to the case and that also has knowledge of it. This is an important process of the paper and hopefully will result in a professionally handled study of the case and the conclusions that are drawn from it.

1.3 Disposition

In this paper I will discuss the human problem in cyber security today and what may be the reasoning for its persistence and for its wide spread. Design and structural decisions that may be of relevance to the human problem when it comes to tech as well as cultural phenomena that may result in furthering the problem. I will present a case that is relevant to the problem at hand. The focus of this paper is how we can make information and cybersecurity better with methods and technology that already exist today. This will be derived through observations from aforementioned cases and expert opinion. This should help to give the reader an idea of what the problem is, why it persists and how we can potentially fix or mitigate it. At the end of this paper there will be a multi faceted conclusion that hopefully can serve as a guide for how to mitigate the human problem within IT and create better systems tailored to people as they are today.

2 Methodology

2.1 Qualitative Methodology

Gathering data on this subject is a rather fine affair. Data on cybersecurity in general is in abundance. However Gathering the nuggets of wisdom that help us learn and get better at what we do is hard to discern from numbers and figures. Although it is a great tool for gathering info about the general situation, I feel that to learn and adapt we must look at relevant situations and learn from others mistakes. Therefore using a qualitative methodology in choosing 'incidents' that are representative of normal use cases here in Norway, I think will serve this paper better. Conducting interviews with people that can describe, in detail, what happened, what the consequences were, what they learned and what they will be doing moving forward, I think, will serve this paper better than pure data and figures. First hand experiences and the interviewees opinions will be an asset for the conclusion of this paper and what I hope to accomplish with it. Talking to an experienced person who knows not only about the incident but about IT and security as well will give the conversation legitimacy and hopefully further the conclusions to a meaningful end.

2.2 prerequisites for Cases

As previously stated the need for care when it comes to choice of cases is really high. Not only will the people that I interview have to have been employed by someone that has had an 'incident' where a system has been compromised in some way and it has to have had some negative effect or at least the potential to have had. They will also have to be informed of that situation, be in some position that directly worked on or has come in contact with that system whilst also having the technical know-how to be able to explain the situation well enough for me to be able to gather the information I need. That technical know-how is quite important. Since I am after the technical detales of what happened, why it happened and more important how it happened. Being able to understand all those points and give technical information about each one is paramount.

Not only will the interviewee be important, the system that will be relevant for that interviewee will also be of high importance. What type of system was compromised? Is that system relevant for today's modern infrastructure? The system needs to be relevant to as many as possible or at least the situation needs to be. This needs to be true for either everyday people, or people working in IT. I am looking for simple mistakes or compromises that had or could have adverse consequences for a large amount of the people involved. Or situations that are highly relevant to people working within the IT sector here in Norway.

2.3 Reliability

The reliability of this paper as a source of information and good practises is a vital point. If this paper is supposed to function as a guide for security or an eye opener for what our systems have to be made for, the selection and reflection of the data used, interviews taken and conclusions gotten from them, has to be meticulous and scrutinous. The main flaw of this paper is its limits. The original plan for this paper was to have 2 interviews to take place from 2 different incidents. However due to the topic at hand it proved quite difficult to find interviewees that were both relevant for the paper and willing to talk about being 'hacked'. It is somewhat of a sore topic for, it seems, everyone. Being compromised in any way is not good press for anyone, and businesses both big and small seem to be very reluctant to talk about it in any capacity. I had hoped that this being a research paper with only anonymous interviewees would lessen the feeling of being scrutinized for their mistakes, but that seems not to be the case. I was aware of this from the start of this paper, people are often unwilling to talk about their mistakes. However I did not think that this was that prevalent. Granted I only reached out to a handful of people that I thought were relevant to the paper, yet the vast majority of answers, if there came answers at all, directly stated that they would not like to talk about this subject matter even when it was anonymized. In the end there was only one interview conducted and although it was very informative and highly relevant to this paper, having only one interviewee will skew any results drawn from that particular case. However I hope that with expert opinion and data, that will somewhat mitigate the one-sidedness of the case study.

What does this mean for the structure of the paper? It means that since there is only one case I will have to do all my comparisons to either global or local statistics or draw them from other sources. And since the scope is defined as inside of Norway I will have to focus on statistics that are representative and relevant to that. However, since later in the paper I will be talking about general practices of the IT sector as a whole, global statistics will be highly relevant especially since the local data reflects what global statistics show to be happening in the world. This also means that I will have to take even greater care in gathering data since that data will not only serve as data but also as a comparison to my case. So interpretation of said data will be very carefully done.

3 Theory

3.1 Modern Security

It's amazing what we have done. What we have created. How in the last 100 years, we have gone from diesel cars and vacuum cleaners to world wide GPS systems, airplanes and the internet. However, with every generation that passes it is expected that the one after them will learn and adapt from their mistakes and blunders. And we do the best we can, however the systems we are taking over in the 21st century are leaps and bounds more complex than the ones left over for generations a 100 years ago.

With the internet, even smaller businesses are dealing with enormous amounts of user data and using systems more complex than anything built in the last 100 years. Think I am exaggerating? Think about the amount of technology it takes just to have a website up and running that is available to the whole world at all times and can take thousands of connections at the same time. All the hardware, the routers, the hosting, the servers, the load balancers and all the infrastructure in addition to just the website itself. Deny it all you want, but what we have built with the world wide web is a complex and amazing jungle of technology.

With complex systems becoming ordinary and how many users normal services have, how we treat cyber security today is very quickly becoming inadequate. But how can something with such levels of complexity be secured? Software development is not just about safe code or solutions. It's about the whole package. From underlying hardware infrastructure to the end user. With vulnerabilities like Meltdown and Specter out there you can no longer assume that if you do your job correctly as a software developer you can be sure that your data is secure since there was something you did not work on that led to a vulnerability somewhere. However that is not the focus of this paper. Hardware faults are hard if not impossible to predict. The layers above hardware are the ones that have a more manageable problem. The human problem.

In this paper I will refer to the human problem quite often. By the human problem I mean the increasing security risk that humans pose to IT-systems. People make mistakes and our systems might not be set up to deal with that. This has been written about often and in depth by cyber security experts. So I will not explain it in depth. However, I will have sources referring to it in the literature section of the paper.[17, 6, 2, 20, 22]

As stated above,we have specialized ourselves. We hire people to do very specific things with a very specific purpose. And as Bruce Lee told us that we should fear the man that has practiced 1 kick a thousand times rather than a thousand kicks one time. When you ask the one kick master how to defend yourself against all sorts of kicking attacks he might not be as useful.

When complex systems are created they are made from many different layers of technology. Defending the contents of that system only gets more and more complex

with each layer. Imagine a castle. The bigger it is, for each new gate and entrance you add you are adding a point of failure to it.

The internet is a scary place. When everything is connected, you are broadcasting data across the world and if you are not prepared you might just invite someone in that was not supposed to be there. Network security is quite literally the most important thing in a system. However it is often thought of. We have experts and teams of people that are very capable of creating and maintaining complex systems. Not always though. When we have as complex systems as we have today, one person's mistake can result in opening the floodgates to all sorts of intrusions and vulnerabilities. Furthermore the most common type of attack is often not accounted for.[20] According to PurpleSec (2019), a cyber security firm based in Washington US, 98% of attacks start with social engineering. Hackers know that the best way to get information or access is through people. In their 2019 cyber security statistics report they go on to say that 92% of malware is sent through email, to people. 21% of all files are not protected and 41% of companies have sensitive files including credit card numbers and health records left willingly unprotected. It's clear that attackers are either very focused on getting through people or relying on badly set up systems, mistakes or just human error.

There is a problem. There is no denying it. Statistics show that we are not maintaining the systems we create in a secure way, and it's starting to cost more to deal with the consequences than we put into security to begin with. Small businesses are getting hit hard these days. They are seen as an easy target. Larger companies and businesses, sure, are a bigger target. However they also have a bigger budget and more people to deal with any attacks or possible vulnerabilities. Small businesses have fewer staff and often no IT or security staff. Relying on hiring contractors that often deliver products that are suited for the mass market and not individually tailored to each customer's needs. [20]Again according to PurpleSec(2019), 48% of breaches are caused by negligent employees or contractors.

So systems are insecure. We know what is causing it. It's us. But is it like that everywhere?

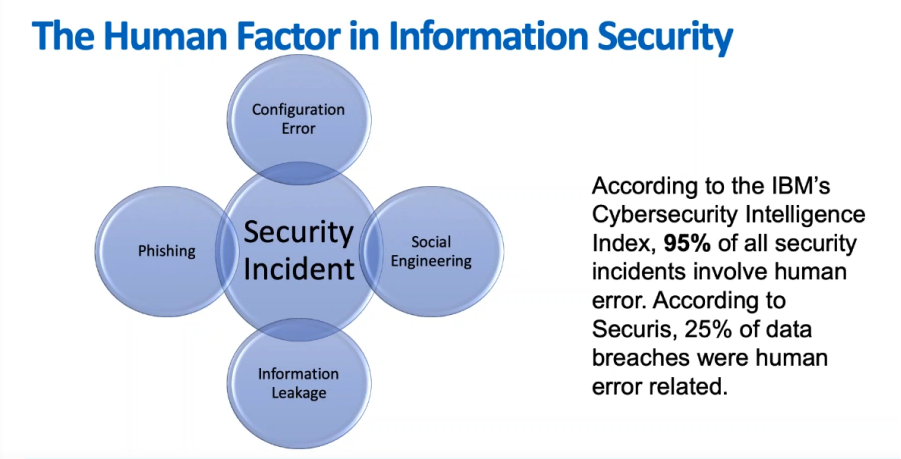

Figure 1: Anthony English: The Role of Human Error in Information Security

[6]Anthony English had a great talk in July of 2019 about the role of human error in cyber security where he outlined the most common points of failure related to security incidents. These are all the most common ways of infiltrating or compromising systems exploiting what Anthony calls the human factor. This is what I refer to as the human problem as I believe that it catches the attention of people better. But in his talk he references the fact that all of these factors are heavily impacted by human error. His talk about the role of human error in information security is highly relevant to this paper and a good amount of the solutions will be based on good practises and recommendations from experts in the field. Such as Anthony English and others.

3.2 Norway's situation

Norway is a country of about 5 million people. [7] According to the UN Development Report in 2018, Norway is at the top of the list of most developed countries. And being a well developed country, we that live here will face the same problems and difficulties that the rest of the developed world will. Norway might be very well developed but it is not above the rest of the world when it comes to security. Statistisk Sentralbyrå is an institution that is in charge of official statistics for the Norwegian government. They have a tool online that is very helpful in gathering statistics about all sorts of things. However, I will be using it to get a better look at the situation in Norway and if the global statistics are representative here as well.

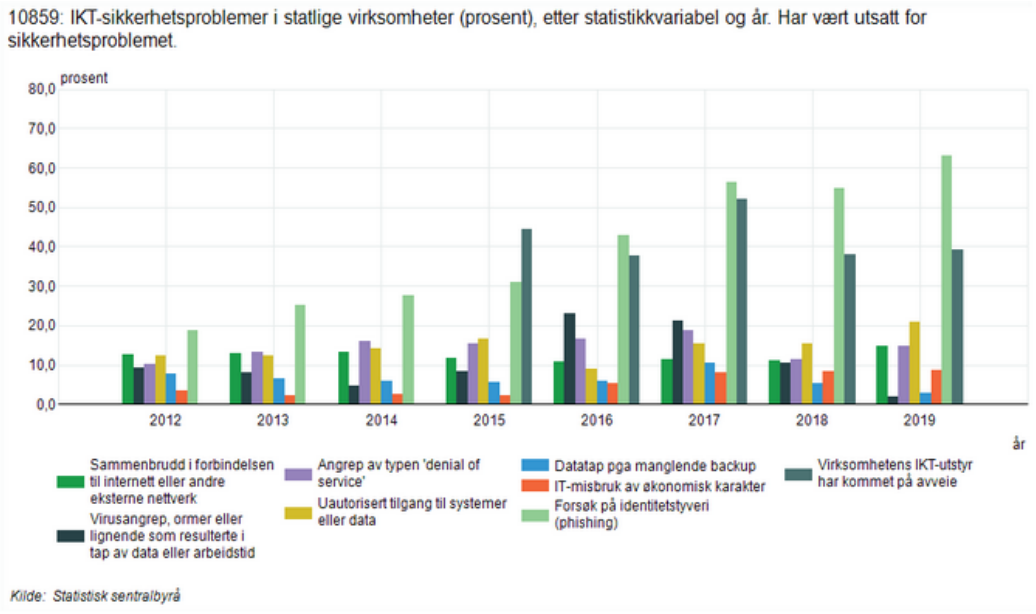

Figure 2: Businesses that have experienced IT security breaches. By year, type in percentages

Pictured above is a graph representing a percentage of Norwegian government entities that have had some sort of breach since 2012. You can see that the percentage of entities that experience breaches of some sort generally are increasing over time, but very slightly. However, the green line sees a general trend upwards. With it being

higher than the previous year in almost every case except in 2018.

Attackers here, like everywhere else, are realising that human error is way easier to predict than the complex systems they are trying to get into. So to see that phishing attempts are trending upwards is no surprise. Exploiting people's trust or naivety is an obvious route to take when trying to get access or information. We are constantly building solutions that are capable of detecting and acting on vulnerabilities and flaws but they are all pointed at the technical side of things. Which is absolutely necessary and should continue and in many ways hinders human error from causing breaches. However there could be something that we need to divert some of our attention to.

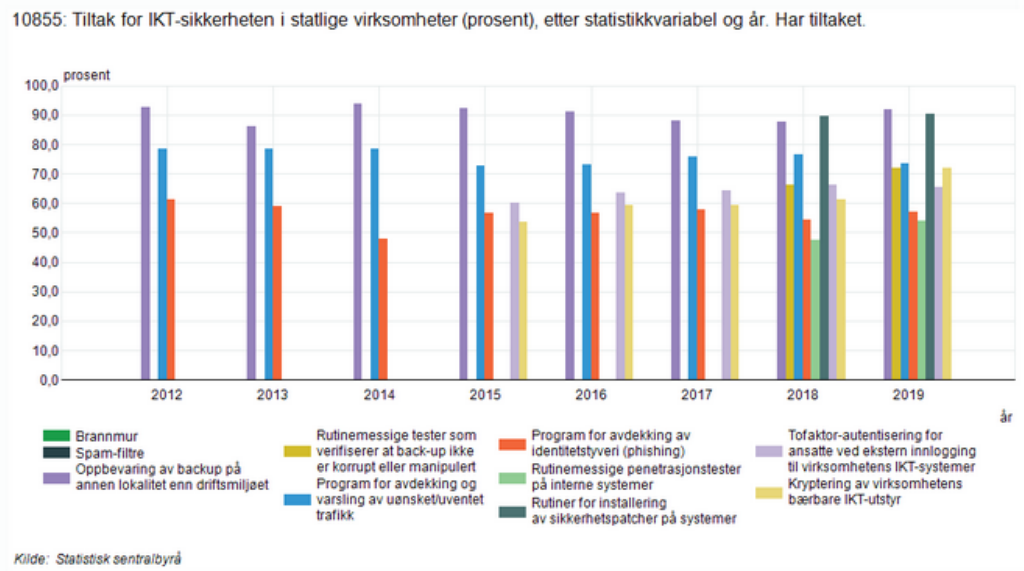

Figure 3: Businesses that have implemented security measures. By year, measures in percentages

Pictured above are preventative measures of government entities in the same time span. There is a trend of increased security measures, yes, but only in the last 2 - 4 years. General safe practises like encryption were not being used until 2015, and only by 50% of businesses. Encryption is one of the most common tools for securing data across the ever increasing size of our systems. These are good changes though. It shows that awareness of the problem is increasing and some measures are being taken. However the focus of this awareness is not at the right point I would say. The way we have set up our systems is flawed. Flawed in the sense that we are not focusing on what is causing the most damage; people. We cannot measure the security of a system by what the system contains. At least not yet. The perfect system does not exist where a person can do everything wrong and the system predicts every bad decision and corrects it. The matrix is not real and the robot uprising has not happened, yet. So we can implement all the security features we want, and we should, but it will not stop "us" from finding a way to mess things up. So what are we doing? We know this problem exists. I am not the first person to talk about this. There are multitudes of people and organisations that are experts in cyber security and how to create and maintain systems in a way that does not leave data exposed. Yet the problem persists, more systems are compromised and more data is leaked.

3.3 Modern systems, old problems

We do a lot of things with technology today. It has come to the point where an online presence is almost a necessity for every business out there. And with technological need comes technological complexity. On-prem servers are almost a thing of the past these days so we have elected a new route to be the future for nearly every type of business. The Cloud. [4] According to this Forbes article about cloud infrastructure (2018) based on a report from Cloud Foundry [8], they estimated that 81% workloads will be in the cloud by 2020. What was in the past looked at as "normal" infrastructure is slowly going away. Yet the place where the most problems persists stays the same. We are getting better and better at defending ourselves but that knowledge does not span the entire business sector. We need to start looking at our data as a part of "the system" as a whole. It does not matter if your data is secure on one server if it is insecure on another. Herd immunity is the concept of the group staying safe from disease by everyone participating in behaviour that keeps the group safe. If one person does something to comprise their safety it affects the whole group. The new technologies we are incorporating in our systems are all well and good. However we need to pull the rest of businesses with us. Keep the group safe by keeping everyone safe. It is no longer an optional thing in business.

3.4 Social Engineering

[1] "One of the most severe practical threats to the confidentiality of information is that the attacker will extract it directly, from people who are authorized to access it, by telling some plausible untruth. This attack is known as social engineering" This was written in the book Security engineering By Ross J. Anderson back in 2001. Since then there have been 2 new versions to come out, one in 2008 and one now in 2020, both fleshing out the details about social engineering ,samongst other things, and what impact it has on the security landscape. The first book talks about how social engineering was first viewed as something insurance investigators would do, they would call hospitals trying to get to information by pretending to be certain patients doctors to get privileged information about said patients treatments and medical history. Since 2001 this type of attack has gone much farther than just the health sector. As PurpleSec [20] stated 98% of cyber attacks use some form for social engineering. Attackers have realized that people are a much more reliable way into systems than waiting for some technical vulnerability. But why are our modern systems failing in this regard? How can we have all this progress and yet user data is flying off the shelves on the dark-net. [5] Well we have been marching on as fast as we can evolving our technology to do amazing things. However we may not have thought this through. We as people are often quick to overlook human nature. We often look at ourselves as we think we should be rather than how we are right now. Humans are idealistic, we look to the future and dream big, and that is not a bad thing. We might, however, want to think about being a little more realistic and analytical when it comes to the technology we use and will use in the future. What are the principles we should go by when creating systems? The statistics show that

the way we act and behave is a vulnerability. As we are now, we are too easily tricked, manipulated and breached. So we might want to think about how we deal with that. Both technologically but also through learning, evolving and becoming smarter and better than we are today.

3.5 Smart system solutions

System architecture and development have come an astronomically long way in the last 20 years. Since the internet spread to all corners of the world like wildfire through dry grass, all our tech is now connected whether that is good or bad. And as a result we now have to think differently. We have to design and develop for the modern world. As the importance of our data grows at a steady pace so does the need for our systems to secure that data as well as the need for smart systems that can either predict or counteract human error.

[12] In 2017, arguably the biggest cyber attack in the history of the world happened. In June of 2017, an update server for a tax-preparation software sent out an update. This software was used by a very large part of the Ukrainian population and many companies that filed taxes in Ukraine used it. That update contained what would later be called 'NotPetya'. A crypto virus that encrypts at the MBR(master boot record) level or a machine, rendering it and the data on it useless without the encryption key. NotPetya was not only destructive for those whose machines got infected, it was also very good at spreading. Starting in Ukraine it ended up with reported infections in France, Germany, Italy, Poland, the United Kingdom, and the United States, though the majority of affected machines were in Ukraine. This attack had a total cost of around 1.2 billion dollars. Maersk, a Danish shipping company that handles a lot of shipping around the EU, was hit the hardest, with their estimate of around 670 million dollars in losses.

How could this happen? A system was set up in a way where attackers could take advantage. Where one person's laziness and decision to not secure a backup server could lead to these consequences. The software company was later found to not have updated their servers since 2013 in addition to using unsafe practices, enabling a backdoor to be created and exploited by the attackers. This should not have happened. So we need to be smarter. Smarter than who? Smarter than us. We need to start planning for this. I am not talking about these major hacks, although we definitely should keep them in mind, we need to think about how we can create and design systems where this would not be possible. Think proactive, this would not have happened or at least not happened in the way that it did, if that one mistake hadn't been made.

3.6 Standards and procedures

Not only does the future call for better technology when it comes to security, we too must evolve. The way we use and manage these technologies must evolve with them. With new technology comes new responsibilities to secure them. In this digital ecosystem it has become more and if we leave a weak link exposed the whole thing will be compromised. The standards with which we judge each link in the chain of data needs to match the seriousness of the potential breach of data of the whole system. Setting your own standards higher than just what is required by law or policy needs to be the norm. Confidence in your own, high, security standards is better than aimless compliance to a minimum requirement by law. Many companies fall into the trap of using the minimum required effort of guarding user data and suffer breaches because of that and most often those breaches come at the cost of those who's data that is. If there is ever a place for shame within IT is is reserved for those who willingly do not put in the effort to secure user data in meaningful way. Because they know it will not have a substantial effect on their bottom line. Do not let yourself become a statistic for bad security. [19]

3.7 Open Source

The open-source movement within technology has been slowly but steadily growing in popularity. Both within the consumer and enterprise space. The logical conclusions that come from complete and transparent access to the solutions and software you implement are undeniably simple and ring true for many people. How can you trust something you do not know how works? The open-source movement would answer that you cannot. There is an argument to be had that the big companies that have years of experience developing and managing software have earned their trust with those years of service and reliability. Yet whenever there is a security breach, people that are impacted often do not know in time to save their data or patch their systems. Whereas with open source software, everything is known. It's out there, you can look at the source code yourself. [9] That is however the crux of the problem. If you are in control of what you implement 100% you are also the one responsible and the one to blame if there is a breach. Using open source you have all the control, all of the responsibility for checking if there is a vulnerability and all of the blame if you did not do so well enough. This is often enough for companies. To be able to say that securing the systems they 'rent' or use is someone else's job and not their responsibility is very tempting for many.

3.8 Culture

With the Agile Manifesto coming in 2001, software and infrastructure development has undergone big changes. We have changed the way we view the technology we build and how we build it. With complex planning and commitment to the end user the product has become a bargaining chip in the starting process of development.

The time it takes to make something being shorter is something the end user always wants yet might not be the way he will get what he wants by the time he wants it. Not giving the developers enough time and then enduring the inevitable delays is often the outcome of such planning. However, delays may even be preferable to what comes with cutting corners and taking shortcuts. Safety is often what is brushed aside when rushing through the development process.

It is becoming widely accepted and expected even, that crunches, rushes, sprints and other words used for 'working hard over a short period of time'. However those terms have started to lose their meaning as companies start extending the crunch time and overworking their employees. [21, 18, 3] A number of mainly gaming companies have been accused of, simply put, trying to demand more work be done in a shorter amount of time than usually expected for that amount of work, and calling it crunch time. Hurrying is the enemy of security. Especially when the people doing the hurrying are tired and overworked. As my interviewee stated, people often make mistakes when they are trying to go fast. Thinking "I just want to do this quickly". And who can blame them when our corporate culture is to demand things for the least amount of money using the least amount of time.

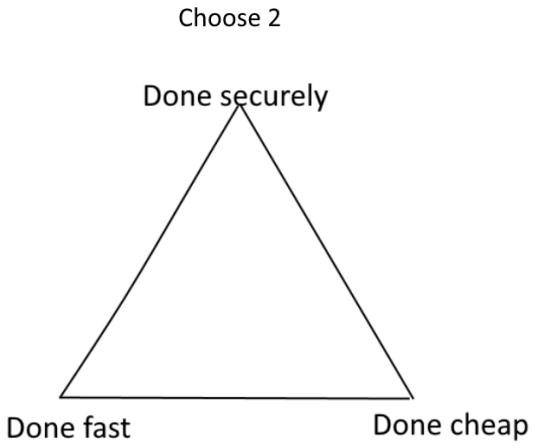

Figure 4: Chose 2

Another big part of the culture today is shame. Shame is a powerful thing. It can make people do things they would normally not, just to spare themselves the moderate amount of shame. Companies are even worse. People often don't have the time and resources to silence those that are about to cause them some shame. But companies often have nothing but time and resources. There is a certain shame connected to IT-security. Shame can be a force of good. It can push people and companies to be better or to fix something when they are in the wrong and have done something bad. Not using good practice, being negligent with data or putting users at risk. Negligence when it comes to security warrants shame. However the kind of shame that is displayed today can be very harmful. Shame that renders you unable to come to logical conclusions and that clouds your judgment that even hinders you from learning from an incident or even fixing the problem can indeed be very harmful. I feel that I witnessed this when trying to find cases for this paper. Talking about security issues is a no go for these companies. Even acknowledging that there is a problem with their systems is not something they do. Often the people who find issues with something get threatened with lawsuits.

Luckily this is not as widespread as it may seem sometimes. Bug-Bounty programs are gaining in popularity and the interest for them both from the companies that get very thorough testing done by people outside of their company and some good press for being open about security and the IT community as a whole. Openness about security issues seemingly helps fight them. However, bug bounty programs are usually just for those companies that are a bit higher up on the corporate scale. Heavy weights with experience and lots of funds under their belt. Medium to small companies that do not have the funds to throw into bug-bounty programs, how can they benefit from this? Well they could at least not bite the hand that tells them about the bug in their software or threaten to sue the persons that tell them about the security flaw in their systems, the person that basically did free investigative work into their systems for them. Being open about security and willing to acknowledge flaws and able to take criticism. To look at what is presented to them so they can analyse, learn and eventually fix the problem.

4 Case

As previously stated, this will be the only case presented in this paper. However it is highly relevant to the topic at hand and will serve as a good example of modern infrastructure and its complexities. All of the statements pertaining to the incident in question in this chapter is either from my interviewee or cited otherwise.

4.1 PAAS

The case I ended up with, after a lot of searching, I think is very representative of the paper as a whole. It fits the prerequisites quite well. The case in question is from a company in the government sector here in Norway. I got in contact with the IT-director of said company through my own network and had heard of the incident in question through him. The most important factor of this case is that the company in question uses a large commercial cloud service as their main infrastructure service. PAAS services or 'platforms as a service' are getting more and more popular within large scale infrastructures. [8] According to Cloudfoundry's survey (2018) on cloud based technology, 50% of companies were using cloud based services and 60% apps and IT-system being developed for cloud based systems. A 13% increase from the last survey just 6 months earlier. So rapid growth is to be expected within this section of the industry. Digitalisation of older systems is a part of that picture. With companies seeking to transfer older systems onto newer infrastructure with minimum effort.

PAAS infrastructure is fundamentally different from 'normal' infrastructure with on prem servers and resources. When you are working on the cloud you are fundamentally 'always online' everything that you do is on a server somewhere that is connected to the world wide web and is not just local to your own network. There are of course security measures in place to make sure that not just anyone can access your private resources, but it is a thing to keep in mind when working with this type of infrastructure.

4.2 What Happened

The incident this company experienced is rooted in the way their infrastructure is set up, so it's highly relevant. What happened was that an employee created a resource to test that was very vulnerable in different ways. What we need to know first is that in this type of infrastructure it's a normal part of the job to create 'resources' within the cloud infrastructure. These can be almost anything. These have basic groups such as compute storage and memory. You can set these resources up to do basically anything you want, just like traditional servers. However unlike with traditional on-prem servers this one is online on the internet rather than being only local behind your own firewall where no one should be able to find it. The main thing about this incident is that as their system was set up is that you can create resources within their system that does not use their own subdomain or certificates. This means that

the resource created gets an assigned subdomain-name and you keep the subdomain before the first dot of the resource URL until the resource is taken down. By doing this you have no control over IP-restrictions, it's all maintained by the cloud service provider. The cloud service provider has a standard implementation for these types of resources. Spinning up these types of resources is not an unusual thing to do.

And the interviewee's team is no exception. Creating resources to test some functionality or compatibility is a part of their daily rutine. However when you are working with online resources you have to think about what that means. It means that everyone can potentially see it. It's on the internet. So As soon as this resource is created, that endpoint is scanned by someone and put on a list. This has been proven time and time again. There are many different ways to scan for different things on the internet and hacker groups are very familiar with them. [10]

You could make these resources with authentication and normal security measures on it. But then you are going a bit further. But since this is what they do, this is normal. As scary as it sounds to create resources with externally exposed endpoints to the internet, that is what they do when they work with a cloud service such as this. Using such services securely is about the security measures you add on top of this structure. And therein lies the main problem.

According to my interviewee the main mistake he sees people do is that they think in the old way of 'test' and 'dev'. "Hey it's only dev/test, it's not prod". Thinking that just because you have not told anyone that the resource is up they don't know it's out there. But you don't have to tell anyone. They already know. The first kind of danger listed by my interviewee is that secrets have been exposed by the resource. If it is not secured properly, is accessible to someone who knows where to look and it has some kind of sensitive information then it is definitely a risk. And even though there is no user data or secrets on there, the people that see it might get insight into their systems or be able to reverse engineer some files. This really exposes them to this threat. The second danger is that if you now want that resource to go to production, you put up protections and you use a domain you own, but in dev and test you still just have the default subdomain the provider gave you.

Example : company-resource-type.cloudserviceprovider.io

Then you just have to rely on that system to work. And let's say there is no data breach. This is a test resource. Someone needs to test it, right? These types of resources get passed around in chats and emails all of the time. You send the URL to the resource that could be anything, but let's say a web-app. People test it, it works or it doesn't, it does not matter. There are two things that happen after this. Eventually this resource will be deleted or moved, even if you obfuscate the old resource and hide it behind the new endpoint, there is still a link. There is a link somewhere to the old resource that does not exist or is not there anymore. And those links now lead to nowhere. And what is nowhere on the internet? It's free real estate.

Anyone can pick up a link that goes nowhere and set up something that the link then leads to. And this is very dangerous. This link, that most likely is still in someone's

bookmarks, email or chat history, is now a link to anyone else's resource. It could do anything. "You could land on a site that just hacks you. Tries to install something or execute code" says my interviewee. "And although that is not as likely with modern and up to date machines, it still is a possibility. What is a more likely situation is a phishing attack. And this even happened to us not so long ago". A phishing attack trying to get logon credentials had been sent around in his workplace not so long ago, causing some password resets. Phishing attacks are more popular these days, as you can do a lot more with credentials than ever before. And usually the goal is to get even more logins with the already acquired one. Causing a cascade effect usually trying its way up the ladder looking for credentials with higher value. Many things have been tried when creating these traps but the simplest way according to my interviewee is just to create a login portal that somewhat resembles the login portal of the service that the specific company uses. In this case the login portal of the cloud provider would make most sense. "People are prompted with this login request multiple times a day. Why should this one time be any different? This is easy, and hackers are so capable these days that any one of them could put this together in an hour and already have it linked up with the URL from the old resource. And even with our team, if it looks like a weird link, they click on it anyways because people are getting so used to clicking on links. You do it so often that the one that sticks out, you don't really think about it." By creating this link and sharing it around you have created the danger for this type of attack and left your system vulnerable.

As mentioned earlier, this is not out of the norm, creating resources that is. So there must be a way of doing this without endangering the system. I asked my interviewee what this person should have done. There must be a way to do this properly. Well there is. He said that there were two main things done wrong here. Firstly, you need to set up principles and procedures that make it so that you can never do this in this manner, exposing endpoints that we have no control over and can not safeguard properly. Always use your own domain. Make it a part of the procedure to always use your own domain so it is not even possible to do it the other way. This can and should be automated. Furthermore, "dev and test" should never be an excuse! If something is worth testing its worth protecting. Also if it has to be external facing you have to make sure that it is very clearly defined what kind of authentication method and security measures you are implementing. So there can be no confusion about it. Even if it's internal you should "ALWAYS HAVE AUTHENTICATION!"

"Not using our domain and certificates is one and having no standard and stated authentication is the other. This combination is what leads to this vulnerability." As soon as you start thinking like this it forces people to start to think that this is the "minimum security level". Some people will think it's hard to do every time you want to spin up a resource so they will automate it. It doesn't really matter how you do it as long it gets implemented as a part of the procedure. The infrastructure needs to be a part of the pipeline. Infrastructure design will be better for it because the minimum requirement will be a part of the default. So set the minimum high!

This is a fairly specific vulnerability for a very specific system. However there are certain similarities we can draw from this case to basically all IT or infrastructure based systems. So we have a system that has users that are capable of causing great

harm to a system with tools and resources given to them by said system. And isn't that the core of the problem?

I asked my interviewee who discovered this vulnerability. His answer was simply that he had gotten sent a link to this resource and noticed that it did not follow the security standards they aspire to and was a potential threat. But what are the consequences? Not what could happen but what did. According to him, for them in this case there was no data in this particular resource. It was just a dummy with no connection to our network and talking only to an API that only had dummy data. An internal security check on the resource revealing that there was nothing compromising with this particular resource. So they made an example out of it. Blew it somewhat out of proportions for the devs so they could understand what could have happened. However they now have to park this resource forever. And create a subscription with their vendor that is called "DO NOT DELETE". They can never delete it because that opens up the potential threat. So no real harm done, only the potential for one.

Such a complicated system with many people working on it. Can any one of them do this? Well no. Like most places this workplace has a permission system that does not allow just anyone with a work ID to do this. Not even every dev can do this. You have to be a senior infrastructure architect to be able to confirm pull requests for resources. So not everyone can do this but this shows that even though permission structures are in place, even senior staff can make mistakes and configure something wrong.

But such a complicated system that has a lot of people working on it must have some systems in place that should help prevent this sort of vulnerability? There are indeed systems and protocols in place, however there is a mentality of "it's only a prototype" or they say " I am just gonna test this real quick" and this happens when people are trying to do things fast. "The technology we use for product development is called terraform. A part of that process is called a 'plan' in terraform. There it checks against policies that we have created. We can create policies that look for those sorts of things. But in theory if you have the credentials to you could override that. What it should be is that the policy check is a bare minimum requirement. If you go below them you are not allowed to deploy anything. Even if it's "only dev or test". It has to become an automated part of the pipeline. People should not be doing this. And when someone misconfigures something it will fail in 'plan' just like if there was a typo somewhere. An integrated part of the system."

What will be done after this though? If nothing changes, there is no progress. "We will be enforcing our policies even harder moving forward. We are also using our cloud provider to help with this. Having them run our policy checks on a different level and pinging anything that does not fit."

If we do not learn from our mistakes we will only repeat them. Having processes in place that deal with these kinds of situations is paramount to keeping our systems safe. So I enquired my interviewee about if there were any procedures in the yearly routine that he thinks are especially helpful in maintaining the safety of their infrastructure. The answer to that was simply, code-review on pull requests. "It puts pressure on

people to do things right. If the plan spits out an error it is no big deal but having another person there makes you feel like you have to impress. That's just how we are. So preventative measures like that are key. Things like pen-testing and such are more just for show and to say that we have done it. But most problems arise from within, so working on preventing those problems before we have a problem is a key mindset to have and will in the long term have a greater yield for security. And technology such as infrastructure as code has made this easier. To be able to review infrastructure setup and not just code or programs."

This is a new and complicated system. It has many facets and nuances that create a big picture when all put together. I personally have heard the sentiment of "we can't expect everyone to know every single detail about the system they are working on." So I asked my interviewee about this. When working with a team on a system that can be considered complex and large, is it unreasonable to expect your team to know the ins and outs of your system, to know how the infrastructure has an effect on development and how development has to take the system into consideration.

Well he did not think so. "Knowing how the system you are working on works is an integral part of maintaining the safety of the system. Especially when modern systems have become so much more involved than before. It's no longer just boxes in the basement that the on prem sysadmin sets up. It is a part of development and as such it is important that everyone is aware of what we are running on, how it works and what we have to do to make it secure. He says it's really important for everyone to know the systems they work on. It's not unreasonable at all unless it is an incredibly large system. You need to segregate modules of course where certain people know more about parts of the system. The 2 pizza rule is a good example. If a team is getting so big that you would need to order 2 pizzas for them you have too many people working on one thing or the thing you are working on is big enough to be split into segments with its own team working on it. And then if your product is so big that it needs two teams you have to behave differently. Then you have to start thinking about infrastructure and how what the teams are working on is going to fit together into one cohesive product. Then each team also needs to know what the other is doing. So you are always in the situation where people have to know what is going on. Spotify is a good example. They changed their app from one product to different modules for each section of the app. So it can release smaller updates more often. They have to know what the other teams are doing so they do not release something that does not work or fit together. It's never enough to have just one person on the top who knows the structure. Otherwise the whole thing will be out of sync."

So my interviewee's problem was not the biggest of deals for him. More of a good example for his team to look at and see how they can do better. But what if the thing that is wrong is not something you have direct control over? 3rd party services and products are very common in the industry. According to PurpleSec's [20] research mistakes made by or compromised 3rd party software is the second highest cause for security breaches with 41% of cases of compromised data. With the only cause above it being negligence. So what are small businesses to do? If they can not do the job themselves or be at risk and 3rd party services being so high in the statistics for

breaches.

I asked my interviewee about this. Firstly, if they themselves use a lot of 3rd party services or software and if they have experienced any problems with those types of services. Turns out yes, they do, and yes they have also had problems with 3rd party services. "You better be careful", he said. "We are working on a framework to review not only the product itself but also the provider. Just today we had a small confrontation with a provider, not because they did anything, we just don't trust them. And that is really hard. And we have been compromised one time that I remember, because of a third party. It was not that bad, but it could have been much worse. That example was, everything is supposed to have single sign on. And when you buy third party tools for something like office or something similar, it's either code that you run locally or it's talking to a server somewhere (adding a point of failure). Then you have to do a security review. Is there going to be data going off our systems? No, OK then we have to ask three more times, are there any possible connections from our system to theirs and if there is we need to see documentation for it. And because of Norwegian law it also has to comply with the 'Data Behandler Avtalen'. We need to know if it handles data. Where it goes and if it's GDPR data we have to go even deeper. And one of those failed. So yes it's hard to completely trust whatever outside party is coming in and doing anything at all really."

So even for a pretty large company such as this it is hard to trust 3rd party providers. How can small businesses stay safe? His answers to this were not that positive to hear. "It's almost hopeless. How are smaller businesses supposed to know who and what to trust. You have to inform yourself. Google it. That is how it is today. How do you know if the app you just installed on your phone is safe? Well you just have to trust Apple or google. So when you are a business you just have to choose a provider that you trust. That can be hard but that is just what you have to do. These big providers do a lot of work that not everyone can keep up with. The usual fail you see in small businesses is that they use services that do not have the same reputation or have not proven themselves to be safe and reputable. If you are going to trust someone or some product with your data you need to make sure that your data will be treated as well as it can be. So being very selective with whom you trust is key. Ask around with other people within your sector. What are they using? Even big companies make mistakes in these matters with many trying to keep as many doors open as possible. Implementing as many solutions as they can, but then their teams have to be bigger with even more infrastructure overhead. That is when corners get cut and shortcuts get taken."

Security is hard and the bigger the system the bigger the problems get. However my interviewee is in a situation that many businesses are in now or will be soon in the future. Using cloud infrastructure. But why? There are obvious pros to cloud computing and infrastructure but I wondered if there are any security specific pro's that my interviewee thinks is pushing both them and the rest of the industry to the cloud. In his opinion security was one of the pro's yes. "It is more complicated but more flexible. The main pro is agility and scalability. To be able to create and manage things, move them around and manipulate the infrastructure is something that is a big plus. But the main factor when it comes to security is the trust factor. Everyone

knows who Microsoft and Amazon are. And we believe that Amazon or Microsoft will do a better job securing our data than someone with a data center in their basement. It's about reputation. Also there are laws in place that say that user data shall not be stored with 'non-reputable' cloud services. It is mandated. And as we all are stuck at home now our infrastructure has proven to be a blessing. It is much easier for us to work from home with the infrastructure we have and the services we are using."

4.3 Summary

As cases go, this one is particularly special in many ways. Both with what what happened and what the consequences were or could have been. This part of the paper will serve as a summary of the case stating only what is necessary for the clarification.

A company in the public sector using infrastructure served from a cloud service provider had a potential security risk. This risk stemmed from an employee with enough clearance to administer and deploy resources, deployed a resource that was not managed or set up correctly. It was set up in a way that did not only not adhere to the security policies of the company but also put the company and its employees' data at risk. Said resource, a resource being a cloud resource of any kind, is connected to the internet. Being connected to the internet leaves it exposed for anyone with the right tools and know-how about internet scanning to be able to see it and even connect to it. This is a quite a serious potential vulnerability already, but the real danger does not stem from the resource being up. The main danger of this kind of resource being put online is that the resource would contain any compromising data that could lead to someone being able to reverse engineer some secrets or data that they are not supposed to be able to see. That being user data or compromising data for the company. The way this resource was put online was without any kind of authentication. So anyone would be able to access it. It was also put online without using a domain that is controlled by the company. Rather using a domain that is automatically given to any resource that is created, using only a bog standard domain that the devs in the company have no control over. And therein lies the main danger of the situation.

Using some kind of authentication would mostly eliminate the problem of any kind of data being leaked. However, no matter if authentication is used it does not eradicate the main problem with this resource. The main problem is that when the resource gets relocated, moved or even deleted, the URL to that resource is not in your control anymore. Inactive URLs are grabbed by the first person to see that they are inactive. And since this is a test resource the link to it will be in emails, chat messages or in someone's bookmarks. Because it is being tested, it is sent around for people to take a look at and test.

So what you end up with is a link with a URL that looks like any other resource from the cloud provider you use, that is not hosting anything you made. It is hosting whatever the next person to take over the URL is hosting. And that in these cases

most likely will be malicious. Hacker groups have scanners just sitting and waiting for these URLs to pop up on their scanners so they can take them over and load something malicious onto them. This is often something that is made for this specific purpose and is already done. So the URL could be populated in minutes after it is released. And it could be anything.

The most likely case is some sort of phishing attacks. And if we look at the statistics that is the most popular type of attack [20]. Within minutes there could be a resource connected to the URL that has a login portal that looks like some standard login portal for a known service whose only purpose is to send that login data to someone causing that login to be compromised. This is not the case for this particular situation though. Since my interviewee saw this resource and what was wrong with it quite quickly after its creation, security analysis was done on the resource. It came up with no data compromised. However, the main consequence from this mistake is that the company will have to create a special subscription for this particular resource that they can never release. Costing the company money and resources maintaining something that now has no use but can never be removed since its removal would cause a security risk to the company.

5 Analysis

The real purpose of this paper is to analyse cases where systems have been put in danger because of the human problem. Try and summarize what happened, what the consequences were and how the company learned from it. To determine how the human factor had an impact on the incident and what that means for future development and work. And finally to surmise potential framework for securing modern systems for the modern day. Both with expert opinions on general cyber security and with my interview I will have good information on how to get to those conclusions. Focusing on what is relevant to both the case itself and a normal work situation here in Norway. To highlight what is relevant to security for people in the IT-sector, showcase these good practises and align them with the examples given to me by my interviewee hopefully divining possible solutions to dealing with the human problem whilst doing so. Learning from others' experience, knowing what the problem is and listening to advice from experts is how to tackle the human problem within IT.

5.1 Culture

The culture of IT is definitely something that has an effect on security in the long run. How software is built and managed is a big part of the human problem. As my interviewee stated numerous times, the concept of hurrying is often what causes lapses in judgments and mistakes. Even the thought that security is not important. The human problem is a phenomenon that we have to combat on its own, however we cannot have the corporate culture working against us in that venture. Creating the situation where people think they have to cut corners or take shortcuts is a security risk in itself and should be avoided at all cost. Humans are bound to make mistakes so let's try and give us less chances to make them. We must create an environment that gives the best outcome. Giving people the knowledge and resources to do the best they can, pushing them in the right direction. In his talk, Anthony English [6] talks about the importance of not making reporting an incident a penalty against the person. Establishing a workplace environment that lets people report issues without fear of losing their job or being ridiculed by coworkers or their boss. Make reporting a natural part of your infrastructure. "This is probably the number 1 issue when it comes to incident management when it comes to human beings, they are always scared to report." - Anthony English

When it comes to our case this is a very important point. Let's say that in the case of this unsafe resource, it was not discovered straight away. Let's say that the person that set up the resource is reading up on the security policy of PAAS resources for testing and realizes that the resource that was created is unsafe in the long run and does not comply with company policy. The resource could even be a threat in the future. How would you want an employee to respond to this? Having him report it immediately without hesitation for worry about repercussions will net better results when managing this issue. Creating an environment that does not disincentivize reporting of incidents is crucial for a quick response time and the integrity of the

systems you are working with.

As to our case, the response to the security issue was not met with immediate punishment for the employee. Instead, they were informed as to why this was not the right way to set up resources even when it's in dev or test environments. This was used as an example both for other employees to help them understand both the gravity of the situation and to create a learning opportunity for everyone. Also to review current policy and more importantly how it is enforced. My interviewee stated that they will indeed be changing how their pull request policy is enforced so that this will not be possible to do in the future.

Another facet of security within IT is the relationship developers have with the end user. New security features are often not so popular. [13] As shown in google's own statistics, less than 10% of google users use 2-factor authentication for their accounts. 2fa is widely viewed as a major step towards basing authentication on more than just your password. Yet it is looked at as a hassle. Just another step for security's sake that breaks the user's workflow. There is also some amount of fear with using it. What if you lose your second factor in your login? Well there are mechanisms in place for that, however that does not render the fear null and void. It is still there corrupting the users mind and making them feel as if security measures like 2fa are unnecessary and in the way. I asked my interviewee what they did to mitigate this and if they sometimes felt like they should or could not roll out certain features because of the users negative reaction. In his experience most of the work involved in rolling out new security measures or policy is in communication and education. Working with the users, informing them of the changes coming and educating them both on how to use it but also why it is important. Furthermore, they got higher ups in the company to both take part in the education but also to publicly support the changes. Leading by example effectively saying "if I can do it then so can you". This work they put into communication and education for the users is monumental to the successful implementation of their security measures. And in doing so they put in the effort that is needed to ensure that the security measures that are put in place have a meaningful impact on security and serve their full potential. A good system in uneducated hands can't function properly. So by doing this they move all the more closer to meaningful change that will hopefully serve to improve security. This is important. The root of the human problem is that we do not take human nature into account. By actively thinking about what we can do to improve not just our systems but the people working on them we take the necessary steps towards securing our systems and data. Systematically implementing this into our policies and procedures is what is needed in the future. Along with some clever technology.

5.2 Trust

Trust is a very delicate thing within IT-security. It's hard to know who you can trust and what is even worse it's hard to know how you can measure trustworthiness . [2] In her talk "Who can we trust' she talks about the shift to a new form of trust. From institutional trust to distributed trust. With the digital age the old form of middle

men with authority holding our trust, centralized and controlled. Now we delegate our trust to platforms, systems, companies and organizations. We have handed off our trust from people to systems. Often ones we don't quite understand. The fact is however, that these systems are only as safe as we build them to be. And in trusting these systems we have to take a leap of faith that the people working on said system have done the best they can with good practise at their side and knowledge in their heads. The problem is when we take that leap blindly. That leap should not come easily and should not be taken without the knowledge that is needed. Do your research.

In the case above I, with the help of my interviewee, described an incident in which trust is indeed a factor of. As he stated people have started to trust the systems we have created. The way we work and use these systems has become predictable. We trust certain things in certain ways and people have started to notice. Social engineering has kept on trending higher and higher because it works. And because we trust certain things. In this particular case it was links. However people have been wary of links for some time now. The usual phishing email with "definitely not a virus, just click this link", people are starting to know not to click on that. In this case however, it is a link not only sent by an employee, but also created by him using the cloud provider they use. That checks a lot of boxes in people's minds. The main thing here is not the link itself. It is the implementation of the resource. People need to be able to trust that their coworkers will not send them malicious links. So the obvious solution is to ban the sending of links entierly ? No that is not the solution.

Trust is important within a business. Not only in the people you are working with but the work they do. Knowing that they will not do things that harm them, the company or its employees. At least not willingly and preferably not by accident either. Knowing that each employee is capable and willing to do their best to deliver good and secure work has to be there. Like Rachel Botsman [2] talks about, we are in the shift between institutional trust and distributed trust and we in IT must do like everyone else and follow along.

Trust is a fundamental concept within IT-security. You could go as far as to say that the field is mostly about knowing who to trust or finding clever ways to check for trustworthiness. We must not fall into the hole that some people get stuck in that "oh the scary security man says to trust no one. Put on your tinfoil hats and follow him into the basement". Security is not about trusting no one, it's just about knowing who to trust. Going back to the case. The thing that went wrong here is that a person did something wrong and that could have led to severe consequences. The solution is not to ban or stop the thing they did. Like my interviewee stated, what is needed here is to educate the person that did said mistake as well as the people around them. Arm the people with the knowledge as to how to be and do better. More systematic solutions to this lapse in security is also needed. However my focus is on the need for education first. Simply patching the problem with policy checks would be ignoring the root cause of the problem. Making sure that the person that did this knows how to do it right the next time is a much more meaningful change. More technical changes are needed, ofcourse. They just cant be the only thing that changes.

5.3 Technology

How we do development is important. How we work together, the way we interact, plan and execute those plans matters. What also matters is what we use. What technology we implement in the development of our systems is important. Especially when it comes to mitigating the known issues of people. How and what can we implement that helps this. Well first we have to define what needs to be mitigated. Talking to my interviewee gave me a chance to learn about a complex workplace, how they manage risk and security. When it comes to security there are some technologies that seem to be more important than others. Using tech that could foresee security risks or even prevent them is a vital standard that needs to be more widely adopted.

Mistakes need to become something that we plan for, not something we do our best to avoid and try not to think about. Expecting that people never make mistakes and then dealing with the consequences when someone eventually makes a mistake is putting our systems and data into unnecessary danger. When talking to my interviewee, he stated several times that proactive security measures are infinitely better than reactive ones. In the case discussed he said that their policies clearly were not strong enough and this should not have been possible. Using technologies such as terraform when it comes to infrastructure allows you to do certain checks that normally would be a little harder to do. They have all the tools they need to create checks and balances in their pipeline that could prevent this sort of thing from happening. Setting up terraform to fail in plan when the pull request does not comply with policy would be one step but that would require stricter policies as well.

Most of the changes they could implement would not be technical in nature but rather logical. Both in their policy structure and when it comes to who can do what within their systems. Making sure that if someone was to create something that is outside of the security norm they would have to both take it up with more than 1 or 2 people and do so very carefully within reason. Closing the possibility for just anyone to do this. However, when it comes to smaller businesses that do not operate in the cloud it is a different matter. Securing systems is on a system to system basis so what should be done, how it should be done and what tech can be implemented depends greatly.

I asked my interviewee about this. What is important for security for even the smaller businesses out there that maybe do not have that much infrastructure to secure? Herd immunity is not only a thing within people but within our data as well. If 5 different services have your data and only 1 of them is compromised, your data is out there. No matter the other 4 services that are securely storing your data. So what is important to update or take into use now more than ever? Well my interviewee stated that implementing some sort of identity confirmation other than just passwords and logins is important. [13] The engineers at google would agree. Moving away from just passwords and going towards multi factor authentication or identity based authentication seems to be something very important and something that can help especially smaller businesses that do not have that much infrastructure but do however have few accounts that are critical to the operation of their business. Securing those accounts with multi factor authentication or with an identity manager

of some sorts could be a huge step towards securing accounts. This does not need to be a hassle for users either. Microsoft's single sign on is used by many companies across different services and logins and with good access control and policies can be very secure, only asking for authentication when some critical circumstance changes, like network or location. Users should be tied to more than just their usernames. This needs to become the norm and maybe even to go further in future. These technologies can however not exist on their own. With great tech comes great responsibility, policies, standards procedures and routines.

5.4 Standards, procedures and routines

All that is stated above can not exist without some framework. Some structure to hold it all in place. Since the human problem is indeed a people problem, it makes sense that most of the solutions would be in the way we deal with people. We do this with contracts, instructions, communication and trust.

Policy is nothing new. Implementing guidelines for people to follow is an essential part of the corporate structure. That does not mean that smaller businesses can't or shouldn't implement them. Showing intent, directing people towards something is important. And choosing what you direct them to is equally important. They need to be specific and there for a reason. In the case above policy was in place that should not have allowed this to happen. However enforcement of that policy was lackluster. Policy that is not enforced is a sign of disregard for security or policies that are redundant, unintuitive or not communicated well enough. Enforcing policy is also about spreading knowledge about why the policy is there in the first place. What purpose it servers and why it should be followed. Policy that is not enforced serves no purpose. Useless in the end. What becomes policy needs to be a matter of standards.

How do we set standards? High that is how. The law sets a standard for security. After GDPR we are a little closer to setting standards even higher for security. Especially when it comes to personal data. However, the law should not set the bar. The law should always be the absolute minimum standard. In the case of my interviewee, their standards are higher than most when it comes to security. Using encryption where it matters and other security measures that are not expected of them, yet they adhere to those standards because they know the consequences of lackluster security. Not only does what we do have to be in order. How we do those things is important as well.

Standardized procedures, knowing how to do certain things. Some things are so important that you want them done a specific way. A good example of this would be spinning up a resource. Something that is done often and is very important is done well and securely. As long as you have policy that dictates what is important, standards that you adhere to and procedures that you follow you have removed a lot of the possibility for something to go wrong. There is one other thing that we need in order to minimize the potential for us to make mistakes. par People work in circles.

Cycles of things we do every day, often without even knowing. Even the general routine of waking up and going to sleep is one of them. The cyclical nature of life. We have to embrace that part of us. Use it to our advantage. Create routines and encourage having security as a part of a daily, weekly, monthly and yearly routine.

Having security as a part of the workflow is yet another way to minimize the human problem and increase the detection rate of any mistakes. You do more checks more often the higher the chance you catch incidents as soon as possible. All of these steps are what is needed in modern security. Give people, not the least amount of opportunities to do things wrong, but the most amount of opportunities to do things right. The right framework is important for security as well as for the people. When talking to my interviewee he expressed that they will have to adjust their framework in the future. Because building that framework is always going to be a work in progress. Especially when working with complex systems. Getting it just right is always going to be a goal that we should strive for and is what will enable us to get closer to more secure systems and workplaces.

5.5 Outcomes

The last link in the chain of safety we need is proper methods for dealing with the fallout when things go wrong. [6] Again in his talk about The Role of Human Error in Information Security Anthony English talks a great deal about what is important for companies to do when it comes to incident response. However I will not be reiterating what he said and would rather recommend you go and watch his talk and read others expert opinions on the matter. However I would like to talk about one point made in his talk.

Figure 5: Anthony English on Incident Management and Distaster Recovery

"One of the most important parts of incident response is also the most often omitted: learning and improving." This is the point that describes a lot of the things I am writing about. The first step in solving a problem is to accept that there is one. The

second then has to be to learn from that problem. There is a lot of focus on 'fixing' problems within IT that are not fixes at all. Just patches to a bigger problem that will eventually resurface if not dealt with. Not only learning why and how something happened but also what were the circumstances that lead to it. What information was missing, what policy was not in place, what standards do you not meet or what routines were not followed. If none of those, what part of the framework is flawed. Looking at the problem as a learning opportunity to get better. Not hiding it in the back of your closet and hope that no one starts looking.

My interviewee and his team went through a lot of effort when it comes to the outcome of his particular incident. To use it to learn about their internal framework and also what their people need in order to do things right. Educating those who need it and communicating what is important. At their core, a lot of technical problems are not so technical at all. Rather just a matter of people not having the information they need or the structure needed for them to acquire that knowledge. Learning both from our own mistakes as well as others is perhaps the most important task IT professionals have today. As we move to a more digital world information and cyber security becomes more and more important. So the safety of our data means the safety of our livelihoods, our work and even ourselves. So making an effort by looking at the problems we face and looking backwards at others that have faced similar issues, is something everyone should have in their routines. Let us not be doomed to repeat the mistakes of the people before us. Let us learn and become greater, safer, smarter.

6 Conclusions

The conclusion of this paper will be in the form of a guide. A short summary of the most important points I have learned from doing the research I did for this paper as well as the interview I conducted. It will hopefully serve as a list of good practices for anyone working in IT or working with technology in general. Since most of the solutions to the problem at hand are of a more humanistic nature rather than a technological one this list will mostly contain structural principles and ideas that will hopefully come in handy.

6.1 Culture

The culture we create in our workplaces and the IT-sector in general is very important. It has an effect on us as people. Ensuring that we do our best to influence that culture in a positive way is paramount.

Companies should strive to create workplaces that foster good results. Not only by having good policy and standards in place, but by setting up a structure that pushes people to do good work. Showing that security is important, communicating why, educating both devs and users alike how to implement practises that reinforce good security. Making changes only after ensuring that the people affected will know that its coming, why, when and finally what it means for them and their workflow.

Not rushing or cutting corners. Because the fallout of security flaws will most likely be monumentally higher than the time cost of doing things right. Not expecting things to get done quicker than normal just because you say it's crunch time. Cost saving measures that rely on less security or test time, if they backfire, can cost way more than what was initially saved.

The workplace needs to give people every incentive to do things right especially when it comes to security. By putting security first and unilaterally including it in everything we do. IT-security should not be an afterthought nor the last step in the process. IT-security should be a methodology. Don't make something and then secure it. You make something secure.

6.2 Trust

Trust is hard. It is hard for both people and systems. So we need to make it as easy for both of them as possible. By using best practices wherever we can within our systems and educating the people. A lot of phishing attacks are unsuccessful at getting through the systems we create to block them. However there are always some that get through and end up in front of someone. And when that happens we need to have done everything we can in preparing that person for it. So when that Nigerian prince needs to send you money or the login link your coworker sent you is

not using an internal domain you will know better. We can't effectively build trust before we educate the people on what they should and should not trust. Mistrust and then confirmation is much preferable to trusting things blindly, so instilling a small amount of paranoia towards things that seem slightly off is a great start.

6.3 Technology

Using technology that improves the potential for security is a must in the modern day. There are a lot of negative attitudes towards security measures that may impede the workflow of users these days. However, the end result is preferable to security breaches that will definitely impede or even halt the users workflow.

Putting time and effort into familiarizing and educating users about the changes beforehand will show a lot of promise when it comes to the reception of those changes. Communicating effectively about why these changes are being made, what type of changes they are, why they are important and how they will impact them will go a long way. Not implementing important security measures because of potential frustration should never be the case. And if there is confusion about some implementations, looking into how you inform users and communicate changes would be a good step to take.

That being said. There are a lot of technologies out there today that are built specifically or that can be used to try and mitigate the human problem. Version control, access control, backup systems, identity managers, credential services and the list goes on and on. We need to start thinking about what we can implement that can lessen the harm of eventual mistakes. Something that can serve as a safety net for people when they misstep.

6.4 P.S.P.R

The need for a framework of good security practises in IT is real. A set of guidelines that helps people do the best they can with the skills and information they have. It is not really about telling people what to do. Rather making a baseline, the minimal requirement. This framework does not need to feel restrictive and overbearing. It is there to guide people towards best practises. It's not there to make sure that nothing goes wrong, it's there to make it as unlikely as possible for things to go wrong.

Your own standards when it comes to IT-security should always be higher than the legal requirement. The law states the minimum requirements. Blindly implementing them with no regard to what suits your use case is never a good recipe for a secure system. Setting your own standards that are relevant to your use case and systems is far more effective both when it comes to implementing features and changes that are useful to you and when it comes to control of what you do implement. You can always go higher, but the law is the bottom.

Certain things require planned procedures. Again it is not about telling people exactly how to do things rather setting the groundwork. Your system might be unique in some way requiring some standard procedures. Making sure that certain things get done a certain way is a good thing to set up. Especially when it comes to incident response or dealing with anything out of the ordinary. When people have room for doubt in vulnerable or stressful situations, mistakes often get made.

Encouraging good routines is a good way to get important security measures into your workflow. Seeing that upper management cares and actively works on security can inspire people down the ladder. Having yearly, quarterly or any time based routines that are managed by the corporate structure can also instill a feeling of importance when it comes to security. If everyone does theirs, we are all a bit safer.

6.5 The Inevitable

Human nature is not about to change in the next few years. We are imperfect beings that make mistakes. We are only human. However our systems are not made entirely with that in mind. We see that the most effective, the most likely and the most used way of getting into systems is through people. The bigger hacks today are very technically sophisticated and the methods of hacking are mind-numbingly intelligent. Yet most of the time, at some stage, they rely on a person to make a mistake, trust someone they should not or just be lazy.

Do not give up hope though. The conclusion is not that human error is inevitable so we might as well give up. It's that human error is inevitable so we might as well plan for it. We need to patch the way we treat people. Instead of just trying to make things people proof, why not try and educate that person while you are at it. People attend work seminars about this and that every day, why not have one of them be about the importance of their actions when it comes to IT-security.

Teaching about the importance of security and making the workplace and etiquette reflect that. Write policy that makes sense and matters, set your standards for security high and make sure everyone knows them and why they are so high, implement good practise and make sure its a part of the standard routine. It costs less to have security a part of the whole process, from the people to the systems they work on, than dealing with the fallout of a serious security incident.

However, there will come a time where something goes wrong. It is as inevitable as human error. So plan for it. Make sure you do the things you can to prevent it, but have a plan ready for when it does happen. Good incident response that makes sure to handle the problem as quick as possible, document what happened and why, what went wrong, and most crucially how to learn from it. Do not make the mistakes of people in the past. And don't let the people of the future repeat yours.

7 Bibliography

References

[1] Ross Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 2001. ISBN: 9781119642831.

[2] Rachel Botsman. *The Biggest Issue in Cybersecurity is Humans, Not Machines*. 2018. URL: https://bit.ly/2LSbJyT. (accessed: April 2020).

[3] Polygon: Colin Campbell. *How Fortnite's success led to months of intense crunch at Epic Games*. 2019. URL: https://bit.ly/3ghkTTz. (accessed: April 2020).

[4] Forbes: Louis Columbus. *83% Of Enterprise Workloads Will Be In The Cloud By 2020*. 2018. URL: https://bit.ly/3gfrIFf. (accessed: April 2020).

[5] cybleinc. *Sensitive Data for Sale*. 2020. URL: https://cybleinc.com/2020/05/20/large-database-of-wishbone-posted-for-sale-online-sensitive-data-for-sale/. (accessed: April 2020).

[6] Anthony English. *The Role of Human Error in Information Security*. 2019. URL: https://bit.ly/2A7saol. (accessed: April 2020).

[7] EU. *Developed Countries list 2020*. URL: https://bit.ly/2VCsq7j. (accessed: April 2020).

[8] Cloud Foundry. *What's Driving Companies to the Cloud*. 2018. URL: https://bit.ly/3bwW6Ix. (accessed: April 2020).

[9] FSF. *The Free Software Foundation*. 2020. URL: https://www.fsf.org/. (accessed: April 2020).

[10] "Robert Graham, Paul McMillan, and Dan Tentler". *"Mass scanning of the internet"*. https://bit.ly/3bC62AB and https://bit.ly/3awnhlj. (accessed: April 2020).

[11] "Robert David Graham". *"Mass scanning tool"*. URL: https://github.com/robertdavidgraham/masscan. (accessed: April 2020).

[12] Wired: Andy Greenberg. *The Untold Story of NotPetya, the Most Devastating Cyberattack in History*. 2018. URL: https://bit.ly/2AOUGIy. (accessed: April 2020).

[13] Google Grzegorz Milka Software Engineer. *Anatomy of Account Takeover*. 2018. URL: https://www.youtube.com/watch?v=W2a4fRalshI. (accessed: April 2020).

[14] Sarah Hospelhorn. *Analyzing Company Reputation After a Data Breach*. 2020. URL: https://www.varonis.com/blog/company-reputation-after-a-data-breach/. (accessed: April 2020).

[15] IDG. *IDG Cloud Computing Survey*. URL: https://bit.ly/2Kw7LLP. (accessed: April 2020).

[16] Ponemon Institute. *The Cost of Cybercrime Study*. 2017.

[17] Kaspersky. *The Human Factor in IT Security: How Employees are Making Businesses Vulnerable from Within*. 2018. URL: https://www.kaspersky.com/blog/the-human-factor-in-it-security/. (accessed: April 2020).

[18] Gamasutra: Jared McCarty. *Crunch Culture Consequences*. 2019. URL: https://bit.ly/3d286me. (accessed: April 2020).

[19] PlainTextOffenders.com. *Plaintext Offenders Master List*. 2020. URL: https://github.com/plaintextoffenders/plaintextoffenders/blob/master/offenders.csv. (accessed: April 2020).

[20] PurpleSec. *Statistics report 2019*. URL: https://bit.ly/351DAFE. (accessed: April 2020).

[21] Kotaku: Jason Schreier. *Inside Rockstar Games' Culture Of Crunch*. 2018. URL: https://bit.ly/2XDEXaz. (accessed: April 2020).

[22] Mahmood Sher-Jan. *Data indicates human error prevailing cause of breaches, incidents*. 2018. URL: https://iapp.org/news/a/data-indicates-human-error-prevailing-cause-of-breaches-incidents/. (accessed: April 2020).

List of Figures