Encryption Recommendation for SMEs

Research Manual

by

Kewin Skuza

Student ID: C00237361

Project Supervisor: Christopher Staff

Date: 25th April 2022

Abstract

Cyber security best practices require data encryption. Not only when we store it, but also in transit. Data encryption became so important that when you ignore it you can get fined if you handle Personal Identifiable Information (PII), as mentioned by General Data Protection Regulation. Not encrypting data can lead to a multitude of attacks. To prevent them, your implementation should include the most up-to-date encryption algorithms. My application aims to classify which data should be encrypted and gives some helpful tips on how to go about it.

Acknowledgements

I would like to thank my project supervisor Christopher Staff for all the help throughout the research process and help with choosing the correct technologies to get the application started.

Table of Contents

Contents

Introduction

Cyber security has become a high priority amongst the IT sector and governmental bodies over the last decade. Data breaches have become an ever present and detrimental occurrence in the tech industry. The EU has brought in strict legislation to force the hands of businesses, creating a necessity for strong security for digital data. Securing data in many cases was left as an afterthought like during the 2014 Yahoo data breach where over 500 million user accounts were stolen [Trautman, L.J. and Ormerod, P.C., 2016]. There is now a massive amount of data which must be parsed through and identified for proper encryption and security.

Automation has become a top priority for much of the IT industry today; productivity can be drastically increased through various automation methods. For data security it is essential to use the correct encryption methods on sensitive data; when companies have collected data and have not correctly stored it poses an issue. It is not feasible to have individual persons working back through massive amounts of data to manually enact the proper security protocols; this is where we can see how indispensable automation is in the modern computer industry. The ability to parse through massive amounts of information and automatically identify Personal Identifiable Information (PII) is essential for both retroactive and current data security. In this document I aim to outline some of the methods which can be used to achieve the automation of this data identification.

Importance of Data Encryption

Data encryption is every system's last line of defence. Ever since the internet was created people have tried to steal customer or company information. No matter how sophisticated your defences are and how much money you have invested in breach countermeasures your company is still vulnerable to zero-day exploits. Zero-day exploits are vulnerabilities in the system that exist, but the company has no idea about. It is simply impossible to secure the whole system. There is always going to be a hole or a back door that can lead into data breaches. Therefore, data encryption is important. Once the breach happens the data is still relatively safe as no one can read it without decrypting it first.

Data encryption has been around for way longer than computers were. The first ever sign of cryptography has surfaced at around 1900 BC in ancient Egypt. The Egyptian scribe simply added some unexpected shapes in between proper hieroglyphics to create a code [Damico, T.M., 2009]. Although it is not cryptography it performs the very same function by confusing the person trying to read it.

The very first proper cipher was created by Julius Caesar in around 100 BC to hide messages sent between his generals in war. Caesar took the alphabet and shifted the letters by 3 spaces, meaning that A became D, B became E and so on. This made sure that the message was scrambled and unreadable by someone who is unaware of the encryption [Damico, T.M., 2009]. It is a very simple cipher that can be easily decrypted. Although the decryption was simple back then most Caesar's enemies were illiterate and just assumed it was written in another unfamiliar language.

The modern version of the Caesar cipher is the rot13 cipher. As the name implies it shifts the alphabet 13 times instead of Caesars 3 [Biswas, M.H., Ali, M.A., Rahman, M. and Sohel, M.M.K., 2019]. By no means it is a secure cipher. It is only used against data that needs to be temporarily hidden. For example, when discussing movie spoilers or puzzle solutions. It is only secure when briefly scrolling past it. It shouldn't be a challenge to decrypt if someone really wants to.

Now almost everything is stored digitally, we need to encrypt data that we think is sensitive. Or at least that was the case until 25th May 2018 when the General Data Protection Regulation (from here known as GDPR) came into effect. Since that day the regulation specifies that all sensitive data or personal information needs to be encrypted. Even data that by itself is not sensitive but can be combined with another piece of information to identify an individual needs to be encrypted. Since this regulation there is another important reason to encrypt your data, the law.

GDPR compliance

General Data Protection Regulation (GDPR) is a regulation that took effect on May 25th, 2018. GDPR is the toughest data protection regulation up to date that brings protection and security to data users around the EU. Even though this regulation is passed only in the EU it still affects companies around the globe. Companies that collect data of any EU citizens fall into this regulation. Handling citizen information is a delicate topic which is why the EU decided to make the fines for breaking such regulation extremely high. Some of these fines can reach up to 20 million euro or 4% of global revenue (depending on which one is higher). To comply with this regulation, we must familiarize ourselves with many aspects of it.

According to article 25 of this regulation any newly developed application must immediately comply with this regulation. This means that throughout the development I must keep GDPR in mind. This mindset is called, as mentioned in article 25, “Data protection by default”.

Somewhere in the early stages of development an information inspection must take place to determine which information should be rounded up to be processed. When considering this action, we need to keep in mind that according to GDPR we must ensure that the bare minimum of it is collected and processed. This ensures that the users’ privacy is not infringed on. They have the right to keep unrelated data private. As the only developer of this application, I’m unable to conduct an audit so I must do a brainstorming session instead to determine which data is used and required for the application to function. We need to detect Personal Identifiable Information to show the company that it must be encrypted. When doing the testing I am sure to use dummy data to be compliant with GDPR. Deciding which data is being stored is not enough to satisfy the rigorous rules outlined by GDPR. We also need to have a legal justification for why we process that data as outlined in article 5. In relation to my application, I require to view the data so it can be protected.

When a data breach takes place the company must show a proof that data protection was considered throughout the whole development process. This can be done by implementing GDPR principles into your documentation while the project is being developed. It is important that the data that is stored is only stored for time that the sole purpose requires.

One of the most important aspects of GDPR is mentioned in article 32. It specifies that “pseudonymization and encryption of personal data” is required to be compliant. To further ensure confidentiality the data must be invisible to non-essential personnel. To ensure integrity, the latest encryption and hashing technology must be used.

A facet of GDPR is related to the people developing the application. Even if technical security is up to par, operational security might still pose a risk. A development policy would be beneficial to develop as well as educate everyone about the importance of data protection. Communication between developers and administrators is inevitable. It is crucial that everyone is educated on how to take personal online security precautions. Some precautions might include but are not limited to the use of a VPN, password security, two-factor authentication, encryption, and awareness about internet phishing campaigns alongside other social engineering schemes. Some of these precautions must be taught to non-technical employees to ensure that the whole chain is secure. As the only member of my development team, I’m going to make sure that no unauthorized persons are viewing my project.

A data protection impact assessment must be conducted throughout the development process to make sure that data sanitization and validation takes place. According to recital 78 of the regulation system administrators adapt and update all system policies up to date. There is a general standard when it comes to the data protection impact assessment.

The comfort and security of the end user is top priority of this regulation. Article 6 of the regulation affirms that the end user has the exclusive right to their information. It gives the end user the ability to keep their personal data to themselves whenever they wish to do so. It is important to inform the user that this is an option. Transparency is key. On the note of transparency, a company must inform all end users within 72 hours that a data breach occurred. If a company fails to do so repercussions will occur.

Article 7 is all about end user consent in relation to their own personal information. If the end user at any stage decides to withdraw their consent on the use of their personal information, their decision must be honored.

Having discussed the various aspects of this regulation I concluded that GDPR is an essential regulation to be implemented into every project. It mainly focuses on privacy and wellbeing of the end user. The end user usually would stand no chance against a corporate giant that does whatever they want with their personal data. GDPR can change that by forcing companies to make the users' personal information as secure as possible using secure cryptographic techniques like block cipher encryption and password hashing. Securing the data is not the only part of this regulation that can be admired. The fact that each user can object to giving their personal data or the ability to withdraw their consent is a huge stone off one's chest. All these aspects are difficult to consider but they are sure to improve the safety of all people involved.

MITRE ATT&CK

The mitre att&ck is a comprehensive and thorough repository that stores information on exploits and cybercrime adversary groups and their known behavior. It was originally created in 2013 by mitre to document all the cyber attacker tactics and their techniques. Since then, it is regularly updated with new exploits and malicious groups [Attack.mitre.org. 2021]. The mitre att&ck repository is trying to understand attacker models, methodologies and how to mitigate them. They want to show how important cyber security is. A surprising number of companies and individuals online are oblivious to what goes on in the background. Mitre is trying to bring that to light and teach everyone how to be safe online. In this section I would like to discuss some mitre att&ck functionalities and some relevant exploits that could have been prevented if an application like mine was used.

TACTICS

The mitre tactics tab explains why a technique would be used. It shows a goal the attacker wants to accomplish [Attack.mitre.org. 2021] . Some tactics include:
Reconnaissance, meaning that all the attacker is trying to do is gather information to plan a future attack.
Executions, which simply means that the attacker tries to execute some malicious code on our system.
There are a few tactics that can be slowed down by an application like mine. In cyber security it's all about increasing the work factor to make it harder for the attacker to get the information they sought after. The tactics include:
Credential Access, the attacker is trying to steal account usernames and passwords. If we encrypt these credentials before storing them we can protect some of our user information.
Collection, the attacker tries to find data of interest to accomplish their goal. Again, if we encrypt our sensitive data we can prevent the attacker from reading that information.
Exfiltration, the attacker tries to gather as much data as they can. The data stolen can lead to further exploits or breach of privacy.

TECHNIQUES

A technique is the means of achieving the tactical goal by performing a specific action. Techniques are the things we need to protect ourselves from. The techniques tab is organised by grouping the different techniques under the tactics [Attack.mitre.org. 2021]. Each technique might include some sub-techniques. Each technique has a page which explains it, gives some examples of procedures and a brief mitigation explanation. For example, if we look into Credential Access we can see “Credentials From Password Stores”. This technique explains that the attacker may look for external databases with username and password information to escalate their privileges giving themselves access to more sensitive data. To give an instance of a procedure we can have a look into NETWIRE which can retrieve passwords from messaging and mail client applications. There are many more that can be found on the mitre att&ck website.

MITIGATIONS

Mitigations are concepts and countermeasures to techniques and sub-techniques used by an attacker. This page has a lot of mitigations that should be incorporated into every system to prevent successful execution of these malicious ideas [Attack.mitre.org. 2021]. You could spend hours looking through this section to make sure that the whole network is secure. I would like to point out one mitigation in particular that my application is going to try to help point out to SMEs. That mitigation is referred to as “Encrypt Sensitive Information”. Just at a first glance you can tell that it stops a good chunk of techniques. Hopefully with this section we can be convinced that data encryption is a very important aspect in system security.

GROUPS

The groups tab goes through the different cybercrime organisations/collaborations that can be attributed to multiple similar exploits/attacks [Attack.mitre.org. 2021]. Sometimes it is difficult to attribute an attack to a single group as they might try to hide it or go by a different name with every exploit. Some groups on the other hand have a distinctive way of doing things which can relay back to them. This page numbers these groups and gives us a little backstory to them. Mainly where they are from and what they are known for. Once we navigate to a specific group page we can view which techniques they used and which software they rely on to perform these exploits.

SOFTWARE

The software tab gives us a list of all software used by the different groups. If the malicious software was used to exploit a system and was found out then it is on mitre att&ck software page [Attack.mitre.org. 2021]. This page gives a nice description of what the software was used for, and which group used it to perform an attack. Once again it is difficult to pair some software as they are used under different names by different groups, Mitre does extensive research to make sure that two pieces of software are the same. After some looking around I found some software that data encryption can help mitigate or at least make the breach less detrimental. These pieces of software are information stealers that can steal username, passwords or even cryptocurrency wallets, depending on the software used. The software I have found include: lokibot, windows credential editor, azorult and carberp. Although this software will still work when we encrypt our data, the information that comes back will be just gibberish. The attacker will not be able to read any of it unless they crack the encryption algorithm, which is not impossible but highly unlikely.

Attacks enabled by unencrypted data

If an organisation decides to not encrypt their information some serious repercussions may take place when a system breach occurs. This is a no brainer since our sensitive data goes into the jungles of the internet where it can be used by anyone for anything. There are a multitude of attacks that can be performed when data is not encrypted securely. In this section I would like to discuss some attack techniques that can occur if we do not encrypt our data with a secure encryption algorithm.

Adversary-in-the-middle

Adversaries that try to perform this attack simply position themselves between two network devices to pass the data received from one of the devices to the other [Daniil Yugoslavskiy, Attack.mitre.org. 2021]. This is dangerous as the adversary can view the information that is being sent. When an adversary sets themselves up like this any traffic going through them may be altered or even fully halted performing a denial-of-service attack. If we ensure that all traffic is encrypted they may not view any information. Also make sure to use the latest authentication protocols alongside all credentials being protected by SSL.
A sub-technique of adversary-in-the-middle is ARP cache poisoning. When a network device does not have a link layer address it will send a broadcast ARP request. The request should make its way to a local network for their ip address to be translated into a MAC address and sent back to the user. An Adversary might wait for such broadcast and reply with their own MAC address. Now the user thinks that it's talking with the internal network while in reality it is talking with the adversary. The mitigation of this attack is the same as its technique.

Automated Collection

Once within a system the attacker can write some scripts or use some commands to automatically look for certain bits of information [Attack.mitre.org. 2021]. If we choose to encrypt our data the attacker will be unable to even search for these files, the encrypted file names will be just gibberish. Even when they get their hands on any of the encrypted files the contents will be more random letters and numbers.

Automated Exfiltration: Traffic Duplication

Some network devices use a feature called traffic mirroring to duplicate some traffic to send to a network analyzer or some kind of monitoring device [Attack.mitre.org. 2021]. An attacker can use this feature to mirror the traffic and send it to a network device they control. This attack can be used in conjunction with Adversary-in-the-middle. Since it might be used in conjunction with that technique, they would share the mitigation.

Data from Cloud Storage Objects

Attackers may be able to access data from cloud storage solutions. Most cloud storage providers issue some security measures [Netskope; Praetorian, Attack.mitre.org. 2021]. Even though the security features are here to use, implementing them incorrectly can lead to data breaches. An attacker might be able to access credit card information, personal information, or the likes if we do not encrypt our data. Cycling encryption keys could be another extra layer of security of cloud storage data.

Data from Configuration Repository

Configuration repositories are used to manage, control, and configure data on remote systems. An attacker might target such a repository to gain access to administration data [Attack.mitre.org. 2021]. Some configuration repositories even allow remote access and administration of devices making this a vulnerability worth securing. There are multiple configuration dumps that are susceptible to such an attack. These include network device configuration dump and management information bases. It is important to configure SNMPv3 (Simple Network Management Protocol version 3) to use the highest level of security to keep our system safe.

Data Manipulation

Attackers might manipulate data by adding things, deleting things, or moving things. This might affect external results [Attack.mitre.org. 2021]. This kind of data manipulation can affect company processes or decision making. There are two types of data manipulation.

Stored data manipulation where the attacker affects data that is at rest.

Transmitting data manipulation where the attacker changes the data en route to another location.

Either way it is important to encrypt the data before we store it and before we send it over a vulnerable channel.

Email Collection

An attacker might have access to or intercept user emails [Swetha Prabakaran, Attack.mitre.org. 2021]. There are a lot of users that send sensitive data like trade secrets or personal information. This kind of data can be valuable to an attacker. Encrypting these bits of sensitive information before sending an email can add another layer of security and mitigate this technique.

Indicator Removal on Host

An attacker might target system generated output like logs or quarantined malware. They can alter or delete this information to either cause damage or hide their further activity on the system [Brad Geesaman; Ed Williams, Trustwave, Attack.mitre.org. 2021]. Just like data manipulation we must encrypt the data both before storing it and before transit. in order to avoid giving the attacker useful feedback.

Steal or Forge Kerberos Tickets

Kerberos is an authentication protocol used to identify if a user is who they are by using kerberos tickets. These tickets are stored in a specific location on the system [Cody Thomas; Tim (Wadhwa-)Brown, Attack.mitre.org. 2021]. If the attacker is able to get this file they can forge or use some of these tickets to authenticate themselves as someone else, opening up a lot of different malicious actions. It is important to enable AES encryption or another secure encryption algorithm in kerberos instead of RC4.

Unsecure Credentials

If the attacker manages to get into the system they might look for some user credentials to or authenticate themselves or elevate their own privileges [Attack.mitre.org. 2021]. In order to store our credentials securely we require encryption. Even when we encrypt our data we still need to store the encryption keys somewhere. It is a bad idea to store them on the same system as the data as they can be found if the attacker is able to get this far. Storing them on separate cryptographic hardware is a far better solution.

Encryption Recommendation

Data encryption must be examined from every direction. If we only encrypt the data that we store, we are still vulnerable to data that is sent over the internet. Threat actors can use packet sniffers like Wireshark to view traffic going through their computer [Wireshark Go Deep., 2021] . Now imagine that a man in the middle attack is being performed and the threat actor is in between you and the destination of your data. They will be able to view everything before you encrypt it in your database or storage solution. Data encryption is important in transit [Encryption in Transit in Google

Cloud | Documentation, 2021]. This means that we encrypt the data before we send it to the destination and decrypt it (if required) once it arrives. This way, if a man in the middle attack is performed and the threat actor acquires our data in transit, they will not be able to view it without decrypting it first. It adds an extra layer of security to our communication.

How to classify and Identify PII

Personal Identifiable Information is the main item that must be encrypted in every system. It doesn't matter what type of system it is and what information it stores. If it falls into the category of PII then it must be secured. Because PII is such a broad term it encompasses a lot of different bits of data. Some data that is not inherently PII might become it when paired with another piece of information. In this section I would like to attempt to define PII as best as I can and propose some solutions to identify it.

PII definition

According to the U.S. General Services Administration PII is a type of information that can be used to distinguish or trace a person either alone or when combined with other data [Gsa.gov. 2021]. This type of information may take many forms and is not linked to any specific category or technology. PII includes but is not limited to: Social Security Number, Name, Address, Phone number and Biometric Data like fingerprints or retinal scans.

Dissecting PII

Certain PII can be identified by its format. PII like credit card details, phone numbers and email addresses can be turned into a simple template that can identify the whole category [Muthusrinivasan, M., Haahr, P. and Cutts, 2013]. For example, a credit card from a specific brand may consist of 15 or 16 digits and have the beginning prefix of 10, 12 or 13. This would mean that a template for that company's credit card number would be {“10, 12, 13”, “15, 16”}. Another example involves email addresses. Each email address has letters or numbers as the email followed by an ‘@‘ sign and then a domain. Designing a template for a word is more challenging. There is no specific format to it. An approach that might be taken could include keywords. Certain data types might include certain words. We can use that to our advantage. By creating these templates, we can identify some bits of PII.

Additional notes on PII

PII includes some fair information practices that should be taken into consideration when handling it.
Collection Limitation - There should be limitations to the amount of data gathered and explicit consent must be given by the owner.
Data Quality - The PII you gather must be relevant to the purpose you decide to use it for. Also, it must be kept up to date and accurate.
Purpose Specification - You need to specify what you are going to use the data for at the time of collection.
Use Limitation - PII gathered should not be made available or used for a different purpose unless when the owner agrees to do so.
Security Safeguards - PII must be secured by safeguards against modification, deletion, use or unauthorized access.
Individual Participation - The owner of PII must be allowed to ask for their data back at any time and their request must be honored. If such a request is denied the owner must have the ability to challenge that decision in the court.
Accountability - The person that processes PII must be held accountable for their actions [McCallister, E., Grance, T. and Scarfone, K.A., 2010].

Issues with PII identification

Since PII is so flexible and any piece of data can become PII with extra input, it is difficult to catch every piece of PII. Making a template for every PII is impossible and teaching a machine learning algorithm to identify it will take gigabytes of positive and negative examples. It is simply not possible to make a reliable system in such a short amount of time with only one developer working on it. I think that I’ll be able to create a system that finds the most common types of PII, but some outliers will be bypassed.

Similar Applications/Systems already developed

In recent years the secure and proper storage of data has become a key issue inside the tech industry. The Introduction of robust cyber security laws is also driving the implementation and development of proper security practices of online data. Many companies have created applications that try to deal with such issues. All applications I have found online rely on the Machine Learning Algorithm approaches. There are only a few small variations in between the software solutions. In this section I would like to define these small differences and correlate them into my application.

Azure Cognitive Service for Language

Azure has created a service that includes PII detection. It uses machine learning to accomplish this goal. The service has a lot of different functionality with the main focus on Natural Language Processing. The ability of detecting PII is a side feature. This service requires the developer to connect to an API via a URL link and a key. You could also use Language Studio which is a UI-based tool that allows you to create your own application using Azure Cognitive Service. The issue with this is that it requires you to choose different resources adding a layer of complexity to using the application. Some users might not know what resources are required to detect the data they are looking for. My system will make an attempt at doing so automatically. [Docs.microsoft.com. 2021]

Amazon Macie

Amazon Macie is another sensitive data detection tool. It is an API just like Azures Cognitive Service meaning you must have some programming knowledge to use it. Macie uses Sensitive data discovery jobs to find PII in your input. Sensitive data discovery jobs require Amazon's storage service called S3. Data inside these S3 buckets can be analyzed and classified. If we want to make this service user friendly we would have to create an application that uses the service in the background. There are some negatives that make it more difficult to use it, like the fact that you need to send the data to this S3 bucket for it to be analyzed. This adds some overhead that will increase the amount of time it takes to detect any PII.[Docs.aws.amazon.com. 2021]

IBM Security Guardium Data Protection

IBMs solution to sensitive data detection is one of the best that I was able to find. It has a clean and intuitive GUI that provides visualizations and metrics. It can also be connected to the whole corporate system to look for out of norm behavior making sure that no data breach occurs from any company sector. IBMs security Guardium also gives us the ability to look through database data directly making it the exact fix for the problem proposed in the brief. As an extra feature this software gives us the ability to encrypt data as we detect it, further decreasing the amount of work required for GDPR compliance. [Ibm.com. 2021]

Conclusion

All the applications mentioned have something that makes it stand out. I will attempt to integrate these positives into one system that secures PII in vulnerable databases while giving helpful recommendations and advice for encryption.

Technologies

Encryption Algorithms

There is a huge variety of encryption algorithms out there. Some are better than others, but all provide a level of additional security. There are hundreds of encryption algorithms to choose from and it is hard to decide which one fits your needs best. There are two algorithms in particular that provide a good level of security and don't take a lot of processing power. The AES algorithm is the industry standard that ensures security when implemented correctly [St Denis, T., 2006 pg. 140]. ECC has more scalability thanks to small key sizes that ensure the same security as large key sizes in AES and RSA [Hankerson, D., Menezes, A.J. and Vanstone, S., 2006 pg. 18].

AES

AES stands for Advanced Encryption Standard. Currently it is the most used encryption algorithm out there. AES is what we call a block cipher, this means that the data is encrypted a 128-bit block at a time using a secure key. The key size can range from 128, 192 to even 256 bits. The longer the key the more secure your system will be. 128-bit keys will suffice but in the near future it will not be as secure. It is a good idea to sacrifice some speed and storage space and use 256-bit keys to make sure that our system is future proof [N-able. 2019].

Modes of Operation

In order for AES to function we need a mode of operation to perform the encryption. There are many modes to choose from depending on your requirements. Some modes of operation include Electronic Codebook (ECB), Cipher Block Chaining (CBC) and Counter (CTR). They vary in security and speed. The fastest mode of operation that doesn't lose on security is Counter Mode (CTR) which turns this slow block cipher into a fast stream cipher [Rogaway, P., 2011].

Stream Cipher

Stream ciphers encrypt data on a digit at a time basis meaning that data is constantly being encrypted. This allows the next digit to begin encryption before the previous finishes. Also, you can use data that has been encrypted before the whole string finishes [Robshaw, M.J., 1995]. This is especially useful for encrypting phone calls or live videos. You can save a lot of time and processing power by using AES with CTR mode.

ECC

Elliptic Curve Cryptography (ECC) uses an elliptic curve over a finite field to encrypt data. The biggest advantage of ECC is the key size. ECC key sizes are way smaller than RSA or AES keys and provide the same amount of security [St Denis, T., 2006]. ECC also uses a concept named trapdoor functions. Trapdoor functions are easy to compute one way but to reverse the computation it's near impossible. An example would include multiplication of two prime numbers. This is an easy task to do for a computer but finding the factors when only given the product of the two numbers is near impossible. These functions grant an additional layer of security. Not many systems use ECC for its complexity. It is much easier to use AES and gain the same amount of security.

Recommendation

In my opinion AES in CTR mode and 256-bit keys is the best choice for a company to use. It provides a level of security that can last for a long time before it needs to be changed and does not require an immense amount of processing power. It is very important that the algorithm is implemented correctly. A good thing to remember is that your system is as secure as your weakest link.

Hashing Algorithms

Hash function is an algorithm that turns a piece of data of any size into a bit stream of the same size. Hash functions are known as one-way functions. This means that you can compute a hash; but given a hash you cannot get the original message back [Buttice, C., 2021]. There are lots of uses for hash functions. The most common use is password storage. Storing passwords in plaintext is a very bad idea. If there is a data breach the password will be free for everyone to view. But if we hash the password before storing the attacker can only get a random string of characters and numbers. There are a lot of hash functions that have been developed since the beginning of computer cryptography. The two most widely known hash functions to exist at the moment are md5 and sha. Some hash functions are no longer safe to use. Hash functions may suffer from collisions and dictionary attacks.

Hash Collisions

Collisions happen when two separate messages can be hashed into the same hash. Collisions are not very common in most hashing algorithms but do happen due to the pigeonhole principle [Rasjid et al, 2017]. Pigeonhole principle means that if we have ‘n’ amount of objects and ‘m’ amount of containers where ‘n’ > ‘m’ then at least one container is going to have multiple objects. This applies to hash functions since the resulting hash is always the same length; there are only so many combinations of letters and numbers that the hash can consist of. On the other hand, there is an infinite combination of words and data that we might need to hash. Based on these facts ‘combinations of data’ > ‘number of unique hashes. This shows that hash collisions are inevitable. Generally speaking, it is near impossible to find these collisions consistently. A good hash function will be somewhat resistant to that.

Dictionary Attacks

A dictionary attack is another vulnerability that hash functions themselves are vulnerable to. In order to perform a dictionary, attack we need to make a lookup table or simply find one online. A lookup table consists of different words and password combinations with the resulting hash beside it [Dhandhania, K., 2017]. When we get our hands on a hash that we wish to decrypt we simply look it up in this dictionary and see the resulting plaintext. It is a very simple attack that is not as common nowadays due to the use of salt.

Hash Salting

Salting is a very important part of hashing. It makes sure that dictionary attacks cannot be performed. The concept of hash salting depends on appending a word or string of characters to our hash and hashing it again. In order to make salting viable we need to use different salt every time we hash a word. This ensures that every time we hash our password the resulting hash will never be the same. The salt we use must also be long and hard to guess. Best practice would be to generate a random string of characters, numbers and symbols, hashing that. Then hashing our password, appending them together and rehashing the whole thing. This makes sure that dictionary attacks are impossible to be performed unless the salt is acquired by the attacker.

MD5 Algorithm

Message-Digest 5 (md5) algorithm was the most popular and widely used hashing algorithm developed. It produces 128-bit hash values. In 2004 a group of researchers found a number of weaknesses in this algorithm. Despite that it was still considered secure until 2005 when a security expert Bruce Schneier declared it ‘broken’ [Wang, X. and Yu, H., 2005]. There are a lot of weaknesses in this algorithm but the most crucial one is a consistent way to find collisions. This makes this hash no longer cryptographically secure. Although it should no longer be used in cryptography it still has a purpose. It can be used as a checksum to verify data integrity.

SHA

The Secure Hashing Algorithm (sha) family is a group of cryptographic hash functions that were published by National Institute of Standards and Technology (NIST). The sha family includes sha0, sha1, sha2 and sha3. At the time of writing this report sha0 and sha1 are the only two no longer secure hash functions from the sha family. Sha2 is further separated into SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256 [Landman, N. en al., 2021]. The most common sha2 function would be SHA-256. It is cryptographically secure and most probably will be for a long time. It is a good idea to use the most secure hash functions up to date to make sure that your system is future proof. My recommendation would be to use sha3-256 as Sha3 functions are more secure than sha2 for the same hash length.

Databases

Databases are created to collect and organize data. There are many types of databases out there and knowing which one is best for your system is crucial. The database types are categorized by what type of data we store, how we query that database and where it resides.

Database Classification

For example, we have a noSQL database which is a category in which the database does not rely on SQL as its dominant access language. Its main advantage is that it does not have to abide by any schema. This means that you can store any unstructured or semi-structured data like graphs. For certain companies this might be an asset while for others a hindrance. noSQL databases are especially useful for storing very large amounts of data and when an admin or application needs to interact with it on the fly.

Database types that are classified by its location include centralized databases and cloud databases. Centralized databases are located, stored and maintained on one site. All users can then connect to it. The opposite of a centralized database would be a distributed database where there are multiple databases distributed among different sites. They are connected remotely. A cloud database is simply a database that is stored and maintained in the cloud, self-explanatory [Matillion.com. 2020]. For small startup businesses a centralized database sounds ideal. It is the easiest to set up and if you need access to it; it's right there.

The ACID transaction property

Some databases have an interesting property acronymed ACID. It is the way a transaction is formed. The acronym stands for Atomicity, Consistency, Isolation and Durability. ACID makes sure that database transactions are performed without a hitch even when a system error occurs. It is a good standard to include but not necessary. To explain ACID, we must go through each section. I'm going to use examples provided by the IBM documentation [Ibm.com. 2017].

Atomicity means that all data exchanges must be performed as a single operation. "For example, when we transfer funds from one account to another, the atomicity property ensures that, if a debit is made successfully from one account, the corresponding credit is made to the other account".

Consistency warrants that data remains in a consistent state from the beginning of the transaction to the end meaning that the values we were given at the start are of the same format at the end. "For example, in an application that transfers funds from one account to another, the consistency property ensures that the total value of funds in both the accounts is the same at the start and end of each transaction".

Isolation implies that the intervening state of a transaction should be secluded and imperceptible to each other. "For example, in an application that transfers funds from one account to another, the isolation property ensures that another transaction sees the transferred funds in one account or the other, but not in both, nor in neither".

Durability guarantees that the transaction that was performed successfully counts as a valid transaction and stays even contingent upon a system failure. "For example, in an application that transfers funds from one account to another, the durability property ensures that the changes made to each account will not be reversed".

MySQL Database Service

MySQL stands for 'My' Structured Query Language where "My" is the name of the co-founder's daughter. It is an open-source relational database management system (RDBMS) [En.wikipedia.org. 2021]. Its main use allows a user to interact with a database using SQL. It is essential if an admin needs to change something or if there is a need to view/lookup any information. It can be used in a General User Interface (GUI) or in a Command Line. The reason why I want to single out this particular service is because that is the one I am most familiar with and will aim to develop an application compatible with it. I will use MySQL databases as a testing ground for my machine learning algorithm.

Issues in relation to the project

Within my research I realized that there are a lot of different types of databases and database languages. It is simply not possible to create a machine learning algorithm that will detect PII for all formats within the time frame given. If I decide to develop this project further I might consider making it compatible with all database formats.

Programming Languages

Python

Python is a high-level open-source programming language that was dispatched in 1991 with the main objective of making it as human readable as plain English [Pythoninstitute.org. 2021]. It can achieve that by implementing complex functions and specific indentation.

I decided to use python to make my application for multiple reasons. Python is one of the most concise programming languages. It is significantly less verbose than other languages such as Java, this makes it less unwieldy and easier to learn in comparison. Python’s execution pace is significantly faster than many of its peers, few languages surpass this speed with C and assembly being notable exceptions.

Python has received widespread use in every part of the IT industry, it is near ubiquitous for scripting and automation. Upon conducting the research for my brief, it quickly became apparent that Python would be the optimal language for my use case. One of Python's many advantages is the overwhelming community support. There are over 100,000 libraries to choose from created by developers and users around the world. The wealth of support available for Python allows you to find effective solutions to issues encountered with relative ease.

Through my research I found two libraries that are going to be essential for my project if I decide to go with teaching my own machine learning algorithm. These include pandas and scikit learn. Pandas is a library that focuses on data manipulation and analysis. In my project I will be required to look at database entries and identify PII. Pandas will allow me to format data and analyse it. Scikit could be useful for machine learning if I choose that approach. If I choose to explore the rule-based method I might look into using piianalyzer to distinguish names from other data. Alternatively, I could look for a pre-trained machine learning algorithm. There is a software development kit named boto3 for AWS Macie that I could use.

React Native

React Native is an open-source UI framework used for development of front-end applications. Its main advantage is the fact that once you make an application you can export it to be compatible on any platform. Both desktop platforms (macOS, Linux, and Windows) and mobile platforms (IOS and Android) are supported, making it universally reliable.[Cross Platform Implementation · React Native, 2021]

The thing that deterred me from using react native was the time limit on completing the assignment. I am unfamiliar with this framework, and learning it, although advantageous, would take away from the amount of time to complete the project. If more time were given I would choose React native over python.

Rule-Based Patterns

Rule-based patterns involve creating a preset that we compare our data to. There are two main types of rule-based patterns: regular expressions and keywords.

Regular expressions (or regex) are sequences of characters that specify a pattern for a word. We can use some techniques to build a standardized way of classifying data [Remove personal information from text with Python, 2021]. For example, if we take an email address into consideration. We can dissect an email into multiple parts. We have the prefix (the starting name of our email), the @ symbol, the domain (gmail, yahoo, hotmail etc.), full stop, and finally an ending (com, ie, pl etc.). Knowing this specific pattern, we can compare it to a word and distinguish whether it is an email address or something else. Regex is not without its weaknesses. Let's say we would like to detect if a field is a home address. When it comes to Irish addresses we have a word separated by commas and potentially “co.” for the county, and an eircode at the end. If that was the case for every address in Ireland, we would have no issue. But everyone types in their address differently, not even mentioning different countries having different formats. With this much variation detecting home addresses with regex will be next to impossible.

We can also use keywords to determine if a chunk of data is something specific. For example, if we have a dataset of all the countries in the world, we can compare them to a field inside of a database. If we have a match, we can estimate that this field is most likely an address field. Keyword matching is not ideal for big datasets as it takes a lot of processing power to compare each and every keyword to every other word.

If I decided to go with strictly a rule-based pattern approach, I would have to put most of my focus into regex as it is the more reliable option of the two.

Machine Learning

Machine learning (ML) has become a widespread solution in data classification only recently. [A Brief History of Machine Learning - DATAVERSITY, 2021] Machine learning algorithms are capable of viewing the data they are presented with and tell the user what it is, all automatically. The machine learning process is long and grueling, but with enough time and resources it is considered the best solution. There are a few steps to perform to have a finished machine learning algorithm. The first step would involve looking for datasets. Datasets are bunches of data that are classified in some kind of way. For example, if we had a list of 1000 home addresses, this would be considered a dataset. A normal dataset is not enough to teach our algorithm. We need both negative and positive examples [Keith McNulty, 2018].

Positive examples involve things that the object we try to classify is. Let’s keep to our example of detecting home addresses. We would have a huge list of actual home addresses in different formats, from different countries. This comprehensive list will be our positive dataset.

Negative examples are things that our data is not. This is a little more difficult to gather. Well, of course random strings of characters are not home addresses. But some coherent words that are included inside an address like county or road might not necessarily be an address. So, sentences with these kinds of words that are not classified as home addresses should be placed in the negative dataset.

When we gather both examples, we can begin teaching our machine learning algorithm. The algorithm at the start will come back with a few types of data. True positives, true negatives, false positives, and false negatives.[Google Developers, 2020]

True positives and true negatives are results that were classified correctly. For example, if we supply “35 Leinster Avenue, North Strand, Dublin” as our input and the algorithm comes back with it being an address, then we have a true positive result. This is indeed an address. Same with the negative. If we supply something that is not an address and the ML algorithm says that it is not an address, we get a true negative result.

While false positives and false negatives are resulting that the ML algorithm is incorrect about. At the beginning we will have a lot of false positives and false negatives. There is no need to worry though. We can use this data to feed it into our datasets. So, when we classify that a piece of information is a false negative, we can place it in the positive dataset. The same goes for false positives; we can place them in the negative dataset.

With time and plenty more datasets, we can teach our ML algorithm to not make mistakes. As mentioned before, it will take a long time to make a ML algorithm that is reliable in its classifications. That's why for this project I will attempt to find a pre-trained ML algorithm to use as part of my classification.

Helpful Libraries

There are a few helpful libraries that I was able to find online that gave me some ideas on how to detect sensitive information. These libraries involve boto3 and urllib 3.

urllib 3

Urllib 3 is a library for Python that allows you to make requests to the internet and get back responses. With these responses you can scrape the page for the html that is on it and do processing on this information. I came up with the idea that if I submit the field into google and look for a specific keyword. For example, if I submit an address into google I would lookout for words like address, rent, or property. If enough of these keywords are found, we can be confident that it is in fact an address. This could be another low hanging fruit detection to utilize.

boto3

Boto3 is a software development kit (SDK) for python made by Amazon to utilize their services using code. I could use Amazon Macie’s pre-trained ML algorithm to check for sensitive information, get back the results and display them to the user in a simple manner. From the documentation I could tell that it is a complex SDK and will require some learning and getting used to the response syntax. Although the learning curve is steeper than I would prefer, it is still a viable way of doing the main chunk of PII detection. With AWS being the lead cloud computing platform learning about its services and how to utilize them would be advantageous to me as a developer.

GUI libraries

Tkinter

Tkinter is one of the most used GUI libraries for python. It is a built-in library so there is no need to install it like other libraries. From my research I was able to find that many features in python GUI frameworks are similar in many regards. Although they differ in certain methods, the overall development is homogenous. When looking for the ideal library for front end development, I was looking for a way to display data clearly using a table of some sort and a way to switch between pages without creating a new window. Tkinter widgets seem to have that functionality. The solution for displaying tables is called treeview. This might be my main way to display the program results. I was also able to find that tkinter has a thing called grid which allows the user to divide the frame into a grid and neatly slot in the different widgets.

Summary and Conclusion

In my research of methods for data identification I have studied various different approaches; some more efficient than others. Machine learning and Rule-Based-Patterns are the two methods I ascertained would be most suitable for data identification.

Machine learning when juxtaposed with Rule-Based-Patterns shows how automated methods can reduce time spent on an issue in magnitudes. Machine learning can automate the entire process of data identification, without requiring direct action from a user. Rule-Based-Patterns will require an individual to manually set parameters for the method to work, making it a great deal more inefficient. Rule-Based-Patterns also suffers from the limits of the user, it is nigh-on impossible to set rules for every piece of potentially sensitive data manually.

Even though both methods have limitations we can still incorporate both into our final solution. The rule-based patterns can catch out some of the low hanging fruit while the machine learning algorithm can look into the less apparent classifications.

Glossary

BC - Before Christ

GDPR - General Data Protection Regulation

EU - European Union

VPN - Virtual Private Network

SME - Small and Medium Enterprise

SSL - Secure Socket Layer

ARP - Address Resolution Protocol

MAC - Media Access Control

SNMPv3 - Simple Network Management Protocol version 3

AES - Advanced Encryption Standard

RC4 - Rivest Cipher 4

PII - Personal Identifiable Information

US - United States

ECC - Elliptic Curve Cryptography

RSA - Rivest–Shamir–Adleman

ECB - Electronic Codebook

CBC - Cipher Block Chaining

MD5 - Message Digest 5

SHA - Secure Hashing Algorithm

NIST - National Institute of Standards and Technology

SQL - Structured Query Language

noSQL - no Structured Query Language

MySQL - My Structured Query Language

ACID - Atomicity, Consistency, Isolation, Durability

RDBMS - Relational Database Management System

GUI - Graphical User Interface

IT - Information Technology

Bibliography

Trautman, L.J. and Ormerod, P.C., 2016. Corporate directors' and officers' cybersecurity standard of care: The Yahoo data breach. Am. UL Rev.66, p.1231. [Accessed 8 November 2021]

Damico, T.M., 2009. A brief history of cryptography. Inquiries Journal, 1(11). [Accessed 9 November 2021]

Biswas, M.H., Ali, M.A., Rahman, M. and Sohel, M.M.K., 2019. A systematic study on classical cryptographic ciphers in order to design a smallest cipher. Int. J. Sci. Res. Publ, 9(12), pp.507-11.

gdpr.eu. 2018. GDPR. [online] Available at: <https://gdpr.eu/tag/gdpr/> [Accessed 10 February 2021].

gdpr.eu. 2018. GDPR. [online] Available at: <https://gdpr.eu/checklist/> [Accessed 10 February 2021].

Attack.mitre.org. 2021. MITRE ATT&CK®. [online] Available at: <https://attack.mitre.org/> [Accessed 12 November 2021].

Attack.mitre.org. 2021. Tactics - Enterprise | MITRE ATT&CK®. [online] Available at: <https://attack.mitre.org/tactics/enterprise/> [Accessed 14 November 2021].

Attack.mitre.org. 2021. Techniques - Enterprise | MITRE ATT&CK®. [online] Available at: <https://attack.mitre.org/techniques/enterprise/> [Accessed 14 November 2021].

Attack.mitre.org. 2021. Mitigations - Enterprise | MITRE ATT&CK®. [online] Available at: <https://attack.mitre.org/mitigations/enterprise/> [Accessed 14 November 2021].

Attack.mitre.org. 2021. Groups | MITRE ATT&CK®. [online] Available at: <https://attack.mitre.org/groups/> [Accessed 14 November 2021].

Attack.mitre.org. 2021. Software | MITRE ATT&CK®. [online] Available at: <https://attack.mitre.org/software/> [Accessed 14 November 2021].

Gsa.gov. 2021. Rules and Policies - Protecting PII - Privacy Act. [online] Available at: <https://www.gsa.gov/reference/gsa-privacy-program/rules-and-policies-protecting-pii-privacy-act> [Accessed 17 November 2021].

Muthusrinivasan, M., Haahr, P. and Cutts, M.D., Google LLC, 2013. Personally identifiable information detection. U.S. Patent 8,561,185.

McCallister, E., Grance, T. and Scarfone, K.A., 2010. Sp 800-122. guide to protecting the confidentiality of personally identifiable information (pii).

N-able. 2019. Advanced Encryption Standard: Understanding AES 256. [online] Available at: <https://www.n-able.com/blog/aes-256-encryption-algorithm> [Accessed 19 November 2021].

St Denis, T., 2006. Cryptography for developers. Elsevier.

Hankerson, D., Menezes, A.J. and Vanstone, S., 2006. Guide to elliptic curve cryptography. Springer Science & Business Media.

Rogaway, P., 2011. Evaluation of some blockcipher modes of operation. Cryptography Research and Evaluation Committees (CRYPTREC) for the Government of Japan.

Robshaw, M.J., 1995. Stream ciphers. RSA Labratories, 25.
Buttice, C., 2021. What is Hashing? - Definition from Techopedia. [online] Techopedia.com. Available at: <https://www.techopedia.com/definition/14316/hashing-cybersecurity> [Accessed 20 November 2021].

Rasjid, Z.E., Soewito, B., Witjaksono, G. and Abdurachman, E., 2017. A review of collisions in cryptographic hash function used in digital forensic tools. Procedia computer science, 116, pp.381-392.

Dhandhania, K., 2017. Dictionary Attacks, Rainbow Table Attacks and how Password Salting defends against them | CommonLounge. [online] Commonlounge.com. Available at: <https://www.commonlounge.com/discussion/2ee3f431a19e4deabe4aa30b43710aa7> [Accessed 22 November 2021].

Wang, X. and Yu, H., 2005, May. How to break MD5 and other hash functions. In Annual international conference on the theory and applications of cryptographic techniques (pp. 19-35). Springer, Berlin, Heidelberg.

Landman, N., Williams, C., Ross, E. and Khim, J., 2021. Secure Hash Algorithms | Brilliant Math & Science. [online] Brilliant.org. Available at: <https://brilliant.org/wiki/secure-hashing-algorithms/> [Accessed 23 November 2021].

Matillion.com. 2020. The Types of Databases (with Examples). [online] Available at: <https://www.matillion.com/resources/blog/the-types-of-databases-with-examples> [Accessed 23 November 2021].

Ibm.com. 2017. IBM Docs. [online] Available at: <https://www.ibm.com/docs/en/cics-ts/5.4?topic=processing-acid-properties-transactions> [Accessed 24 November 2021].

En.wikipedia.org. 2021. MySQL - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/MySQL> [Accessed 24 November 2021].

Pythoninstitute.org. 2021. About Python | Python Institute. [online] Available at: <https://pythoninstitute.org/what-is-python/> [Accessed 25 November 2021].

Docs.microsoft.com. 2021. What is Azure Cognitive Service for Language - Azure Cognitive Services. [online] Available at: <https://docs.microsoft.com/en-us/azure/cognitive-services/language-service/overview> [Accessed 26 November 2021].

Ibm.com. 2021. IBM Security Guardium Data Protection - Overview. [online] Available at: <https://www.ibm.com/products/ibm-guardium-data-protection> [Accessed 29 November 2021].

Docs.aws.amazon.com. 2021. Discovering sensitive data with Amazon Macie - Amazon Macie. [online] Available at: <https://docs.aws.amazon.com/macie/latest/user/data-classification.html> [Accessed 30 November 2021].

Wireshark.org. 2021. Wireshark · Go Deep.. [online] Available at: <https://www.wireshark.org/> [Accessed 2 December 2021].

Google Cloud. 2021. Encryption in Transit in Google Cloud | Documentation. [online] Available at: <https://cloud.google.com/docs/security/encryption-in-transit> [Accessed 2 December 2021].

Reactnative.dev. 2021. Cross Platform Implementation · React Native. [online] Available at: <https://reactnative.dev/architecture/xplat-implementation> [Accessed 5 December 2021].

DATAVERSITY. 2021. A Brief History of Machine Learning - DATAVERSITY. [online] Available at: <https://www.dataversity.net/a-brief-history-of-machine-learning/> [Accessed 10 December 2021].

Google Developers. 2020. Classification: True vs. False and Positive vs. Negative | Machine Learning Crash Course | Google Developers. [online] Available at: <https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative> [Accessed 13 December 2021].

McNulty, K., 2018. How does Machine Learning work?. [online] Medium. Available at: <https://towardsdatascience.com/how-does-machine-learning-work-6dd97f2be46c> [Accessed 12 December 2021].

DECLARATION

*I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.

*I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.

*I have provided a complete bibliography of all works and sources used in the preparation of this submission.

*I understand that failure to comply with the Institute’s regulations governing plagiarism constitutes a serious offence.

Student Name: (Printed) Kewin Skuza

Student Number(s): c00237361

Signature(s): Kewin Skuza

Date: 25th April 2022