AWS Cert Alert
Research Document

Oisín Chelmiah

C00246756

Supervisor:

Dr Keara Barrett

Abstract

For the research part of this project, I'm delved into what Secure Socket Layer (SSL) certificates are, why they are important, and what ways businesses currently manage them. I'm also researched platforms for managing certificates, including cloud architectures, and comparing the benefits of one platform over another. My main aim with this research document is to see how the management of SSL certificates can be improved upon by looking at what software is currently available for certificate management.

Table of Contents

Table of Figures

Introduction

In very basic terms, a certificate is used to prove something. College degrees are a type of certificate that proves a student has studied and is competent in a specific field/industry. In the software world, there are multiple types of certificates used to prove that a device is who they say they are or that a service is doing what it is supposed to do. There are different certificates for web browsing, emails connection, software publishing, and networking.

For the research part of this project, I have examined a particularly important type of certificate, known as an SSL certificate. I delved into how they function, how they are managed, and what tools are available to aid companies manage their certificates.

I also researched into cloud computing platforms. Cloud computing is a practice that has been growing in popularity in recent years and continues to evolve every year. Companies can host websites and other systems utilising cloud infrastructure, and part of hosting any system will involve certificate management.

During my research into cloud platforms, I learned how companies use cloud computing for resource management, so I compared the similarities between certificate management and resource management to see if I could adapt my system to make it a more attractive option over the options that are already available in the certificate management field.

As part of this research document, I also investigated what programming languages I could potentially use throughout my project and decide which one would be the most beneficial, or possibly see if it would be necessary or beneficial to implement multiple languages.

Overview

SSL/TLS Certificates

SSL & TLS

In the early days of the Internet, Web users were susceptible to a range of Man in the Middle (MITM) attacks, such as eavesdropping – where a threat actor can intercept messages between two end devices and see the data that is being transferred – and malleability issues – where a threat actor can physically change data in transit so that data that the destination device receives is not the same as the data that the source device sent. This was because the connection between the source and the destination devices was not secure.

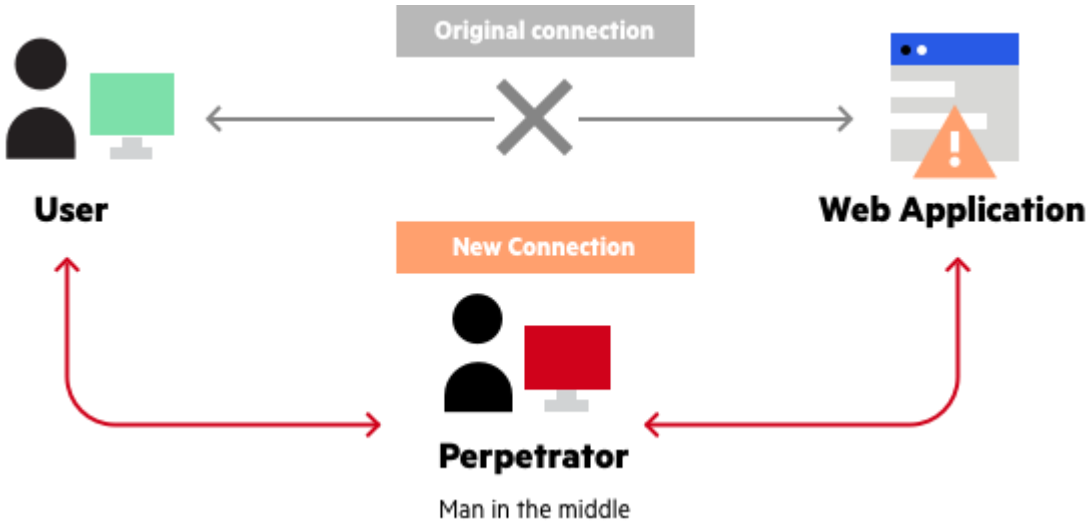

Figure 1 – Example of a MITM attack

In 1994, the Secure Socket Layer (SSL) protocol was created by Netscape Communications to combat such security issues. Netscape created SSL 1.0 and soon after updated their protocol to SSL 2.0 and 3.0 to compete with other emerging security protocols such as Microsoft's Private Communication Technology (PCT). In 1996, the Internet Engineering Task Force (IETF) decided that the development of SSL should be the responsibility of a standards organisation instead of being monopolised by a single corporation. The IETF took over the production and upkeep of the SSL protocol, and in 1999 they officially came out with the Transport Layer Security (TLS) protocol, which was their updated version of the SSL protocol. The upkeep of the TLS protocol is currently the responsibility of the Transport Layer Security Working Group (TLSWG), which is a subdivision of the IETF.

(Thomas, 2000) (Prodromou, 2019)

What is an SSL/TLS Certificate?

SSL certificates are an internet standard for securing a connection between two clients and any data passed between them. SSL certificates are controlled and distributed by public Certificate Authorities (CAs), which I talk about in more detail later on in this document.

Although they are most commonly referred to as SSL certificates in industry, SSL certificates are actually TLS certificates. The SSL protocol is still officially owned by Netscape Communications, whereas the TLS protocol is an open-source standard governed by the international standards organisation IETF.

They function by using encryption algorithms to secure data being passed between two clients. This provides a website with:

- Confidentiality – Ensures only authorised devices can view the data.
- Authentication – Verifies that the device that sent the data is the correct device and is not a threat actor pretending to be that device.
- Integrity – Ensures that the data that is sent from the source device is the same data that is received by the end device.
- Non-Repudiation – Proves that the device that sent the data did in fact send the data.

The two most common types of encryption algorithms used are Rivest–Shamir–Adleman (RSA) algorithms and Elliptic Curve Cryptography (ECC) algorithms.

(DigiCert, 2022) (Thomas, 2000)

TLS Handshake

TLS encryption algorithms work by using an TLS handshake between the client and server addresses to implement their chosen key exchange algorithm. It functions as follows:

- Both the client and the server will exchange details about their TLS configuration. Details exchanged include:
  - TLS versions supported
  - Cipher suites supported
  - Client and server randoms – random strings generated by the client and server used for validation
- The server uses a public and private key which is predetermined by the TLS certification authority.
- The client will read the server's TLS certificate and encrypt a secret message using the server's public key as provided by the server.
- The client then sends back this encrypted message.
- The server decrypts the message using its private key.
- Once connection is established the chosen RSA/ECC algorithm will be used to encrypt any future data that is passed between the client and the server.

(DigiCert, 2022) (Cloudflare, 2022)

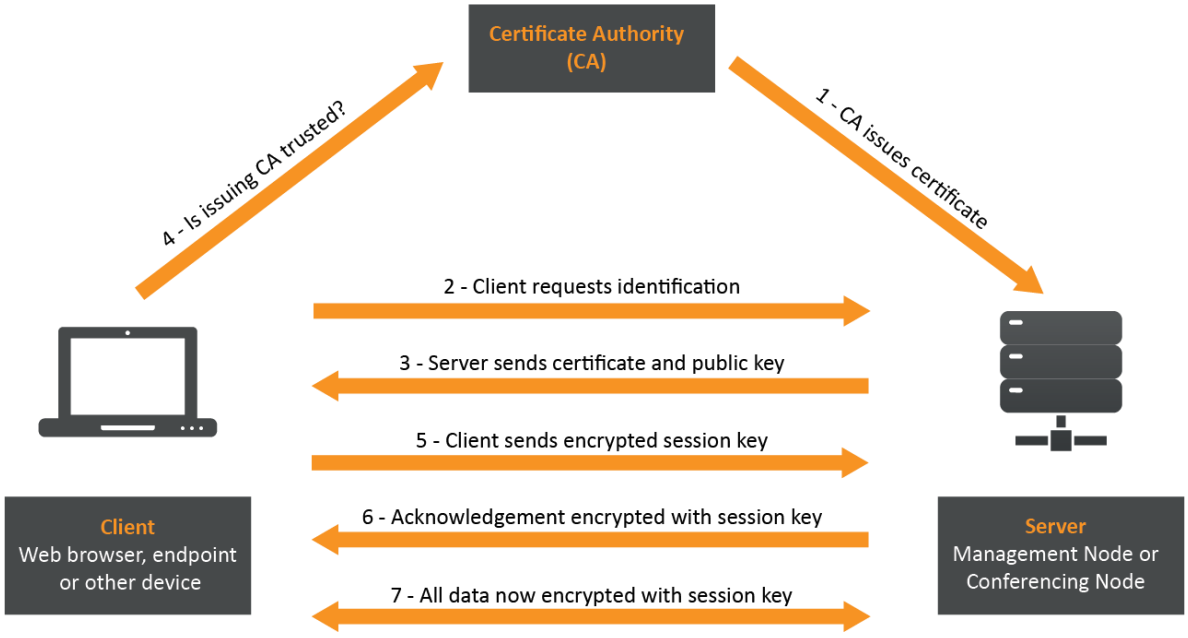

Figure 2 – TLS/SSL Handshake

RSA Algorithm

The RSA was created in 1978 by Ron Rivest, Adi Shamir, and Leonard Adleman. The algorithm works based on the mathematical problem of factorising large numbers.

There are three main steps to the RSA algorithm:

- Key Pair Generation
- Encryption
- Decryption

At the start of the key generation process, both the source device and the destination device decide on two large prime numbers are chosen. The public key is then made up of the multiplication of these two prime numbers along with another integer that cannot be a factor of the above result. The public key is used by the source device to encrypt the data being requested by the destination device.

The private key is generated by using the initially selected prime numbers along with utilising a complex mathematical algorithm known as the extended Euclidean algorithm to create a new pair of integers. The private key is used by the destination device to decrypt the encrypted data that was sent by the source device.

RSA keys are generally 1024 or 2048 bits in length. Smaller RSA keys can be generated but are not recommended since they can be easily deciphered.

(Educative, 2022) (GeeksForGeeks, 2022)

(Zhou & Tang, 2011) (Mahajan & Sachdeva, 2013)

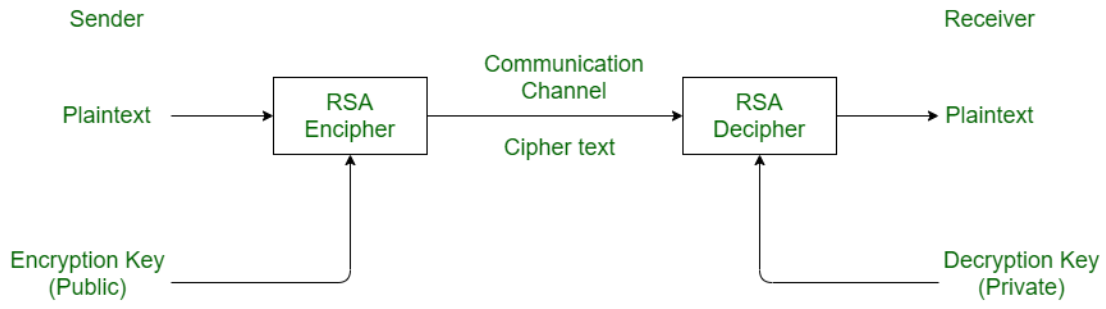

Figure 3 – Encrypting using RSA

ECC Algorithm

The ECC algorithm was first proposed in 1985 by Neal Koblitz and Victor Miller. The algorithm utilises the Elliptic Curve Discrete Logarithm problem. The algorithm works by using an elliptic curve as shown below:

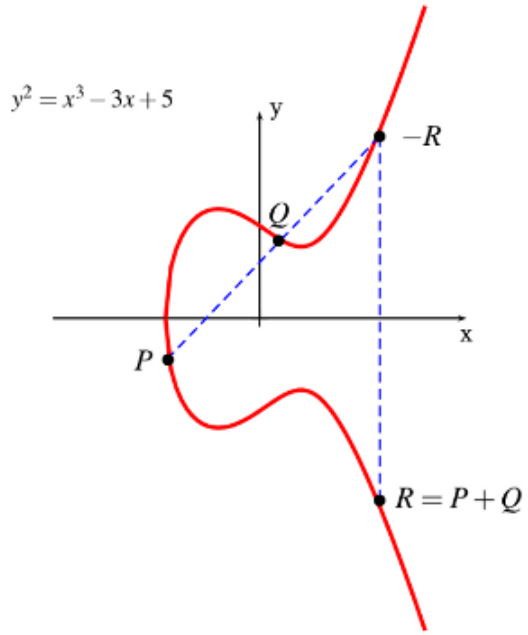

Figure 4 – Elliptic Curve

Before key generation, a fixed point on the curve is decided between the source and destination devices.

Public keys are generated by both the source and the destination devices selecting points on the curve. Private keys are chosen by each device selecting a random integer. Each device's public key is known to the other device, while the private keys are kept secret. Each public key will have and *x* and a *y* value corresponding to a point on the curve.

The cipher text is generated by the source device creating a pair of points on the graph using the destination device's public key.

To decrypt the cipher text, the destination device multiplies the first point by its private key and subtracts the answer from the second point to get the value that was initially encrypted.

(Kapoor, et al., 2008) (Mahto & Yadav, 2017)

RSA vs ECC

Both ECC and RSA encryption algorithms are suitable for use with SSL certificates, with both having different benefits. According to a comparative study between the two algorithms performed by the National Institute of Technology Jamshedpur:

"ECC outperforms RSA regarding operational efficiency and security with lesser parameters" (Mahto & Yadav, 2017)

However, RSA also has its benefits over ECC, such as having faster operations, faster key generation, and faster encryption.

Characteristic	RSA	ECC
Smaller Key Size		✓
Fast Key Generation	✓	
Lower Processing Power	✓	
Fast Encryption	✓	
Fast Decryption		✓
Extra Security		✓
Lower Space Requirement		✓
Faster Transmission with Short Messages	✓	
Faster Transmission with Large Messages		✓

Table 1 – RSA vs ECC

(Kute, et al., 2009) (Kapoor, et al., 2008)

(Mahajan & Sachdeva, 2013) (Mahto & Yadav, 2017)

Certificate Authorities

Before any type of certificate can be used, they have to be acquired from somewhere. This means someone needs to create and distribute these certificates, a role which is taken up by Certificate Authorities (CA). CAs can be split into two types: public and private. Both types will always have a unique root certificate, which is used to digitally sign all certificates that the CA issues. This validates that the certificates the authority issues are:

A. Issued by the CA and not an imposter – since each CA has a unique root certificate and a unique signature; and

B. Secure certificates that are ready to be used and will provide the functionality they are supposed to provide.

(Grubbs, 2022) (KeyFactor, 2022)

Public Certificate Authorities

Public CAs provide certificates for the general public to use. Public CAs are generally trusted companies that have been proven by the public to provide reliable certificates, but governments can also act as CAs too. Public CAs are useful for any public facing product/service. When using public CAs, a company must pay for each individual certificate. Public certificates can be used for encrypting connections to web servers, encrypting emails, and digitally signing documents.

(IBM, 2021)

Private Certificate Authorities

“Where a public CA hands out certificates to anyone who pays, private CAs restrict their certificates to specific people or devices (usually those within the organization).”

(Grubbs, 2022)

Private CAs are internal to a company and create certificates for internal use within a company. These certificates are not available to the public and only function within the organisation. These certificates provide extra security within the organisation’s network. Since these certificates are managed internally to the organisation, the organisation will not have to pay per certificate used, but rather just pay the cost of creating and maintaining their certificate management system.

(Bhardwaj, 2020) (KeyFactor, 2022)

Certificate Management

Once an SSL certificate has been created, not much upkeep is needed. However, it is important to ensure that the certificate has been created properly and assigned to the correct site. Failure to create the initial certificate properly can result in fatal errors for the website associated with the certificate.

All SSL certificates have an expiry date. It is extremely important to keep track of when certificates were created and when they need to be renewed. If a certificate expires without its owner realising, not only will the owner lose the function of the certificate, whatever website the certificate was associated with will become insecure which could result in huge vulnerabilities. Hence, a certificate expiring is in itself a vulnerability, one which could very easily be overlooked.

An incident reported by the Irish Times provides a recent example of how an expired certificate can significantly impact an organization. As per the report, a major incident occurred when the certificate used by operators to communicate with the computer system expired, resulting in the loss of all functionality of the 999 Emergency Call Answering Service and its fallback automated answering system. This incident affected more than 216 callers within an hour and twelve minutes, highlighting that certificate expiry can have a severe impact not only on the organization but also on the general public who rely on its services.

(Foxe, 2022)

Certificate management is often the role of an Identity & Access Management (IAM) team. IAM involves controlling who has access to what resources within a business, and TSL certificates are a crucial resource for any company. It is important to control who can view or edit the TSL certificates, and if the wrong person is accidentally given access to a certificate, it could prove catastrophic for the company.

Certificate Management Tools

Overview

There are some tools out there to aid with certificate management, and in this next section of this document I will be comparing some such documents.

Tool	Publisher	Pricing
SSL Certificate Management and Expiration Monitoring Tool	SolarWinds	€1,353 per year
DigiCert CertCentral® TLS/SSL Manager	DigiCert	€800-€2,225 per year
Key Manager Plus	ManageEngine	€1,420-€2,226 per year
AWS Cert Manager	Amazon Web Services	Free if AWS certificates are used, else charged on a case-by-case basis

Table 2 – Certificate Management Tools

Note: The above prices may vary depending on how many certificates you are managing with each tool.

SolarWinds

Figure 5 – SolarWinds Logo

SolarWinds' certificate management tool is a popular tool for certificate management on an enterprise level. However, their tool only works if your business is also using SolarWinds Server & Application Monitor (SAM).

SAM provides data on the performance of web servers in a company's network as well as monitoring the availability of web services connected to those servers. It provides a lot more than just certificate management, including keeping track of login attempts on webpages, automated scanning of system environments to discover server issues, and managing web server response time and page requests.

SAM includes an alert for expiring TSL certificates; however, it has no way of renewing these certificates or creating new ones. SAM must be deployed on a server machine. SAM does not have an Application Programming Interface (API), meaning security teams are limited on what customisation they can implement on the application.

(SolarWinds, 2022)

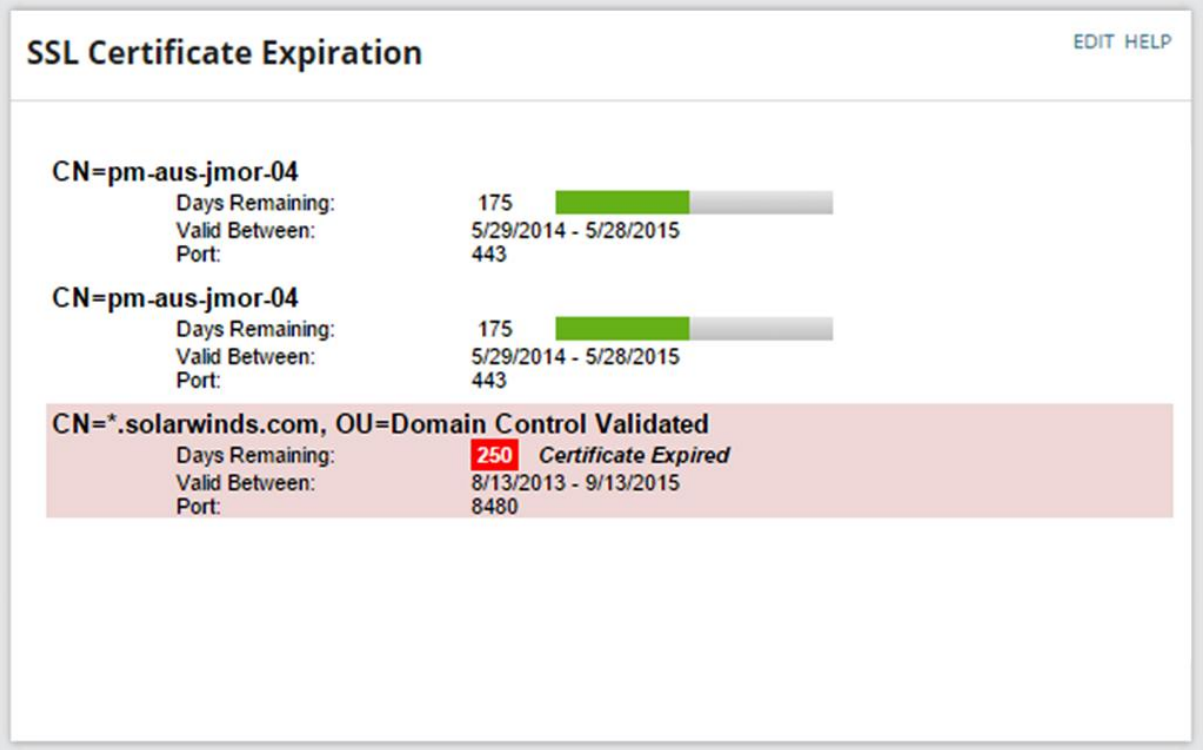

Figure 6 – SolarWinds: SAM

DigiCert

Figure 7 – DigiCert Logo

DigiCert's tool, CertCentral, allows for automated certificate management by providing a consolidated place to install, issue, and inspect TLS certificates. One of the main benefits to using this tool for certificate management is that it can automatically renew TLS certificates when they expire. This can be beneficial if a company knows that it will definitely need to renew a specific certificate that may be crucial for the company's operations, however it could also lead to extra expenses if a certificate that is not being used is being automatically renewed without the company's knowledge.

CertCentral is also compatible with ServiceNow, a tool that many IAM teams use to keep track of access requests within a company. ServiceNow functions by allowing employees to request access to resources within their company. Once a request is made, an associated ticket is made. This ticket contains the details of the access request and will be reviewed by the employee's direct manager and the IAM team. Allowing CertCentral to be integrated with ServiceNow creates a huge incentive for businesses to use the DigiCert tool for certificate management.

(DigiCert, 2022)

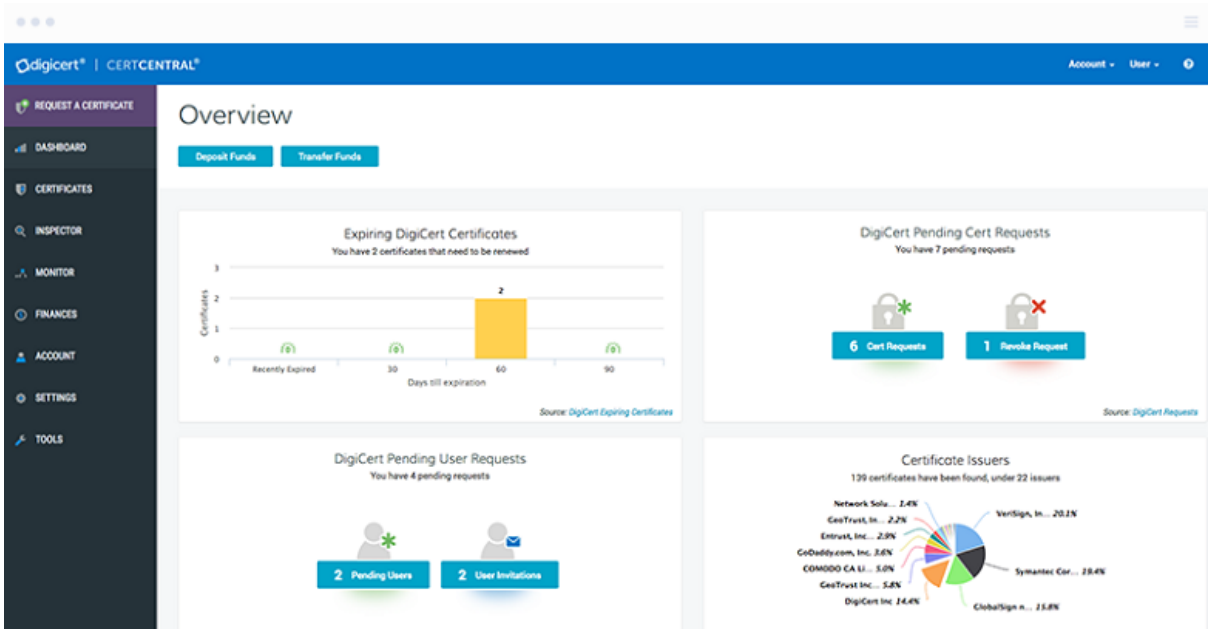

Figure 8 – DigiCert: CertCentral

ManageEngine

Figure 9 – ManageEngine Logo

ManageEngine's tool, Key Manager Plus, works in a similar way to the SolarWinds tool. Once the tool is deployed to a network, it can be configured to consolidate all TSL certificates used within that network, as well as deploy new certificates to specific domains. It does include a certificate expiry alert.

Key Manager Plus also has a Secure Shell (SSH) management system that can be used in conjunction with the TLS certificate management system. This system can be used to manage how a company provides access to certain infrastructure devices, such as servers, routers, or databases. The SSH management part of the tool works by creating and storing SSH keys, which can be given out to user devices on request. This extra functionality is a great incentive for businesses to employ Key Manager Plus in their system.

Another incentive to use the ManageEngine tool is the fact that it is compatible with Microsoft Active Directory (AD), which is what many businesses use for Identity and Access Management (IAM) within their organisation. IAM involves controlling who has access to what services and when they have access to them, and Microsoft AD makes this easier by allowing managers/security teams to assign users to security groups that have pre-configured access assigned to them. Key Manager Plus allows for businesses to keep track of and manage any certificates that may be assigned to specific users or security groups within Microsoft AD.

(ManageEngine, 2022)

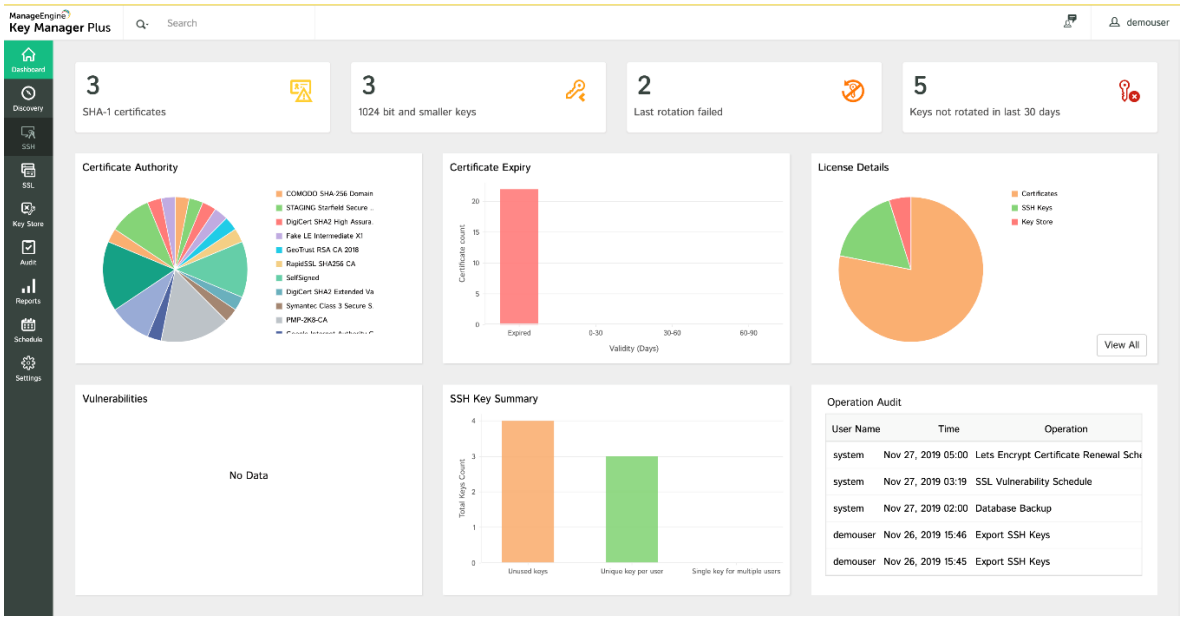

Figure 10 - ManageEngine: Key Manager Plus

AWS

Figure 11 – AWS Logo

Amazon Certificate Manager (ACM) is one of many services that AWS provide. It allows users to create or import certificates and then store the details of those certificates in one place. These details can then be used by other services – as long as the user knows how to properly access them, which can prove to be tricky for new or inexperienced users.

There are two types of certificates that ACM can hold: native certificates that are created within ACM, and imported certificates created externally to ACM. For certificates that are created native to AWS, Amazon has its own encryption key algorithm that it assigns to these certificates. Any certificates that a company wishes to import into AWS from an external source must already have either an RSA or an ECC encryption key associated with it. Imported certificates are not compatible with as many AWS services as natively created certificates are, but ACM can still pass details of external certificates to a lot of other AWS services.

For both native and imported certificates, the details stored within AWS are as follows: the domain names used by the certificate, the expiry date, the public key algorithm, the signature algorithm, and any AWS services that the certificate is compatible with. Certificates managed by ACM are trusted by multiple browsers, including Chrome, Microsoft Edge, Firefox, and Safari.

(Amazon Web Services, 2022)

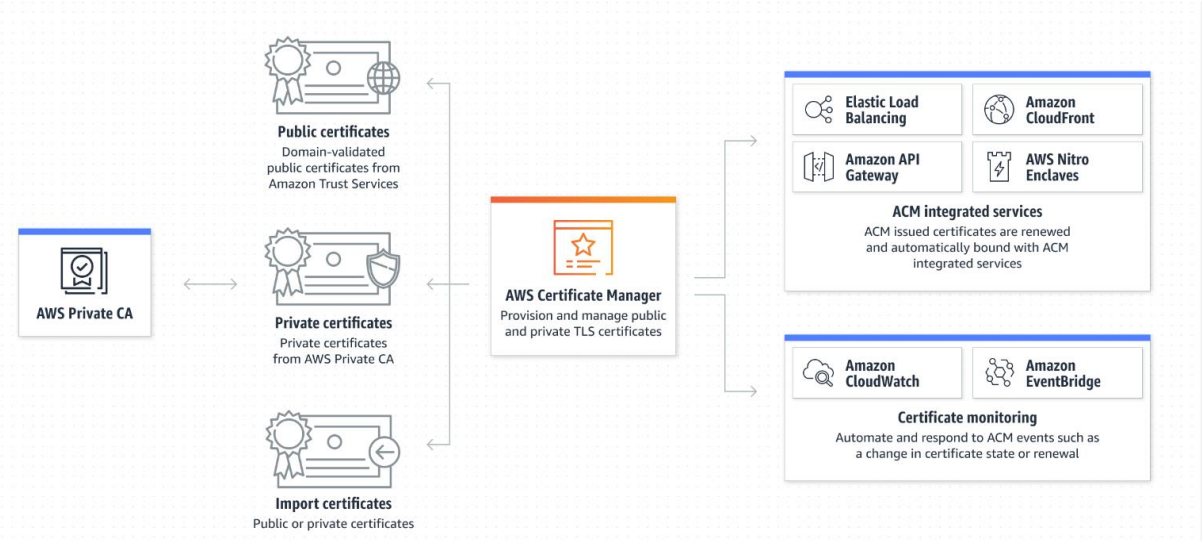

Figure 12 – AWS: ACM

Summary of Management Tools

Although each of the tools described in the above section function similarly, each has extra aspect that sets it apart from the other tools. The SolarWinds tool is included in their SAM tool which provides more data on the servers using the TSL certificates rather than just managing the certificates themselves. CertCentral is compatible with ServiceNow which allows it to be easily integrated into current IAM practices and allows for automatic certificate renewal. ManageEngine's tool has an integrated SSH key management tool which provides extra security by creating a way for IAM teams to manage remote access to servers and other hardware devices. Finally, what sets Amazon Certificate Manager apart from the others is the fact that it is built in as part of a cloud management tool – Amazon Web Services – and as such can reap the benefits of being part of cloud architecture.

While the other companies created their tool as a third-party application that would have to be integrated with other server management tools in order to be effective, AWS designed their certificate management tool to be compatible with their cloud server management system. This means that a company could create their entire system – including an external website, and an internal system with resources and resource management – natively within AWS and without the need for third party services.

ACM only provides basic management of certificates, so I plan to expand on the functionality of this tool within AWS. AWS has an API for all of its services, which allows will allow me to heavily customise the functionality of the tools available to me. One aspect I plan on adding is a consolidated dashboard which will be connected to ACM to allows a security team to view details of all the TSL certificates their company are using. I also plan on using AWS services to create a notification system for expiring certificates to make the management process easier.

(Amazon Web Services, 2022) (ManageEngine, 2022)

(DigiCert, 2022) (SolarWinds, 2022)

Cloud Computing

Overview of Cloud Computing

Cloud computing is a practice that has been growing and evolving in the computing industry. Cloud providers operate by simply letting other companies use their server for a price. As well as allowing the use of their server, cloud providers often add extra functionality and include different services in their cloud packages that may be useful to their customers. These services can include security, analytics, storage, and management tools.

Cloud computing is split into three main categories:

Infrastructure as a Service (IaaS)	**Platform as a Service (PaaS)**	**Software as a Service (SaaS)**
Access to backend hardware needed for running applications/web servers/systems. Includes network devices, storage devices, and virtual servers, all accessed through the cloud. With IaaS, the customer manages their own applications while the cloud provider manages the infrastructure hosting these applications.	PaaS refers to the use of cloud platforms for developing, running, maintaining, and managing platforms. With PaaS, the customer is still in control of their applications, but they have access to development software to help then run the applications as well as the infrastructure to host the applications.	With SaaS, the customer pays for an application that is entirely controlled by the cloud provider. The software is managed and hosted by the cloud provider, so the customer has limited access to customise the functionality of the application.

Table 3 – IaaS, PaaS, & SaaS

Cloud providers will often provide options for customers to use either one of the above types of cloud infrastructure or even mix the three of them depending on their application needs.

(IBM Cloud Education, 2021) (Google, 2022)

(Amazon, 2022) (ProjectPro, 2022)

Resource Management in Cloud Computing

A huge benefit to companies using cloud computing platforms is the resource management aspect these platforms provide. Cloud computing platforms offer a range of services to aid with resource management, including:

- Provisioning – Cloud computing platforms allow users to provision various resources such as virtual machines, storage, networking, and applications on-demand.
- Monitoring – Once resources are provisioned on the cloud, they can be monitored using various tools provided by the cloud platform. These tools allow users to track resource usage, performance, and availability. This enables users to identify any potential issues and take action to optimize their resources.
- Autoscaling – Cloud computing platforms offer autoscaling capabilities that allow resources to be automatically scaled up or down based on demand. This is particularly useful for applications that experience spikes in traffic, as it ensures that the resources can handle the increased load without any downtime or performance issues.
- Optimization – Cloud computing platforms provide various tools and services that enable users to optimize resource usage and costs. These tools allow users to identify and eliminate unnecessary resources, optimize resource usage, and reduce costs.

(Islam, et al., 2016) (Munir & Palaniappan, 2013)

Resource Types

The types of resources that cloud platforms can provide can be categorised into the following sections:

- Computing Resources – Resources that enable users to run applications and workloads on the cloud. These include virtual machines, containers, and serverless computing resources. Virtual machines (VMs) are fully isolated, software-defined environments that can run different operating systems and applications. Containers provide a lightweight, portable way to package applications and their dependencies. Serverless computing resources allow users to run applications without the need to manage servers.
- Storage Resources – Resources that allow users to store and retrieve data on the cloud. These include object storage, block storage, and file storage. Object storage allows users to store and retrieve unstructured data such as images, videos, and documents. Block storage provides raw storage volumes that can be attached to VMs and used as disks. File storage provides shared file systems that can be accessed by multiple VMs.
- Networking Resources – Resources that enable users to connect their resources and applications on the cloud. These include virtual networks, load balancers, and firewalls. Virtual networks allow users to define and manage their own network topology on the cloud. Load balancers distribute incoming traffic across multiple compute resources to ensure high availability and scalability. Firewalls provide network security by filtering incoming and outgoing traffic.
- Database Resources – Resources that enable users to store and manage structured data on the cloud. These include relational databases, NoSQL databases, and in-memory databases. Relational databases provide a structured way to store data in tables with relationships between them. NoSQL databases provide a flexible way to store unstructured or semi-structured data. In-memory databases provide high-performance access to data that is stored in memory.
- Application Resources – Resources that enable users to deploy and manage their applications on the cloud. These include PaaS and SaaS offerings, which are detailed in *Table 3*.

(Islam, et al., 2016) (Alam, et al., 2018)

Threats Cloud Resource Management

As with any computing practice, there are a number of threats that a user faces when deciding to use a cloud platform to manage their resources.

Threat	Description
Data Breaches	Storing sensitive data on a cloud platform can increase the risk of a data breach. If an attacker gains unauthorized access to a cloud account or a cloud service, they may be able to access and steal sensitive data.
Insufficient Access Controls	Misconfigured or insufficient access controls can allow unauthorized access to cloud resources. For example, if an access control policy is not configured properly, an attacker may be able to access sensitive data or resources.
Shared Technology Vulnerabilities	Cloud platforms often rely on shared technology, such as virtualization, to provide compute resources. Vulnerabilities in this shared technology can potentially impact multiple customers on the same cloud platform.
API Vulnerabilities	Cloud platforms provide APIs that allow customers to interact with and manage their resources programmatically. However, if these APIs are not secured properly, attackers may be able to access and manipulate resources.
Account Hijacking	Attackers may attempt to hijack a cloud account by stealing login credentials or exploiting vulnerabilities in authentication mechanisms. Once an attacker gains access to a cloud account, they can potentially access and manipulate all of the resources associated with that account.
Orphaned Resources	Orphaned resources in the cloud refer to resources that are no longer in use or needed but have not been properly deleted or removed. They can occur when a user forgets to delete a resource after it is no longer needed, or when a resource is left behind due to an error or misconfiguration. Orphaned resources can pose a security risk because they can potentially be accessed and exploited by unauthorized users. Additionally, these resources can also lead to increased costs because cloud providers typically charge for the usage of resources, even if they are no longer in use.

Table 4 – Threats with Cloud Resource Management

(Alam, et al., 2018) (Bhan, et al., 2019)

(Dabrowsk & Mills, 2011)

Orphaned Resources

Orphaned resources are an interesting vulnerability that often go unnoticed by cloud security managers. As explained in *Table 4,* orphaned resources in the cloud are resources that are not actively being used but are still present in a user's cloud environment. These resources can include virtual machines, storage volumes, network interfaces, or other cloud resources.

Orphaned resources can occur for several reasons, such as when a user forgets to delete a resource after it is no longer needed, or when a resource is left behind due to an error or misconfiguration. This can happen, for example, when a user creates a temporary resource for testing purposes, but forgets to delete it after the testing is complete. Orphaned resources can also occur when a user accidentally deletes a resource that is still in use, which can result in a resource being left behind without an associated user or purpose.

The presence of orphaned resources can lead to security risks and increased costs for cloud users. Orphaned resources can be exploited by attackers to gain access to sensitive data or to launch attacks on other resources in the cloud environment. In addition, cloud providers typically charge users for the usage of resources, even if those resources are not actively being used. This means that the presence of orphaned resources can lead to unnecessary costs for cloud users.

To mitigate the risks associated with orphaned resources, it is important for cloud users to regularly monitor their cloud environments and identify any resources that are no longer needed. Users should also have procedures in place to ensure that resources are properly deleted or removed when they are no longer needed.

(Alam, et al., 2018) (Dabrowsk & Mills, 2011)

Cloud Computing Platforms

Amazon Web Services

Figure 13 – AWS Logo

Costs	Benefits	Limitations
• Costs $1 to set up an account which is reimbursed. • AWS Free Tier allows access to 60 services. Some services are always free, and others provide 3 week or 12-month free trials. • Charges based on what services are being used and how much data an account is using.	• Most services provided out of any cloud service provider. • Developer functionality allows for customisation. • Certification and training provided. • Allows third party tools to be integrated. • Easy to read documentation. • IaaS, PaaS, SaaS	• Technical support fee • Can be easy to make mistakes when managing a system using the AWS interface. • Easy to be overwhelmed with the amount of services available and not use them efficiently.

Table 5 – AWS Analysis

In 2002, Amazon created Amazon Web Services to handle its online retail operations. After realising that these services could be useful for other businesses to manage their internal systems, Amazon released AWS as one of the first cloud computing solutions in 2006.

AWS is comprised of over 200 services which allow for a multitude of serverless computing solutions for a range of business types. AWS provide storage solutions, database hosts, network management, governance & compliance, web & mobile development, notification systems, monitoring systems, and many more services.

It costs $1 to set up an AWS account, but this charge is reimbursed after 3 days. The charge is in place to aid with validating new accounts as well as help prevent denial-of-service attacks occurring. A user must have and AWS account to avail of AWS services. A company can have multiple AWS accounts. AWS charges on a subscription operated basis based on what services an account is using and how much storage they are using within each service.

AWS provides a free tier of services as well. This free tier includes services that are always free of charge, three-week free trials of services, and twelve-month free trials. Businesses can use these trials to see if a service would be of benefit to their infrastructure before committing to a subscription to that service.

To aid companies with the customisation of each AWS account, AWS has an API as well as a library – known as the Boto3 library – to allow customers to create unique functions within

their accounts. With the Boto3 library, users can write code to access AWS resources within their account, pass details of resources or the resources themselves between different AWS services, and integrate different services together with ease.

(Amazon, 2022) (TechTarget, 2022)

(Mufti, et al., 2020) (ProjectPro, 2022)

Microsoft Azure

Figure 14 – Microsoft Azure Logo

Costs	Benefits	Limitations
• No free tier products. • Storage charges. • Base charge per service which can increase depending on service usage. • Extra data transfer costs.	• Compatibility with Microsoft products. • Expandable. • Help and support easily available. • IaaS, PaaS	• Comparatively hard to use. • More expensive than competitive cloud platforms. • Documentation hard to read. • Requires expertise to use efficiently. • Limited functionality when it comes to tailoring the infrastructure's design.

Table 6 – Microsoft Azure Analysis

Microsoft Azure is another popular cloud computing service, although it functions very differently to AWS. Since Azure is designed by Microsoft, it is compatible with other Microsoft tools such as Microsoft Office and Microsoft Server Manager. The integration of Azure and Server Manager means that businesses can take a hybrid approach to their server management; Server Manager can be used to manage company owned servers and Azure can be used to manage their cloud servers.

Like AWS, Azure has an API to allow customisation within each Azure account. However, whereas AWS has the one library that contains commands to interact with services, Azure has multiple libraries for each of its services.

Azure and AWS are very similar in a lot of aspects, but AWS has designed their system in a way that is easier for inexperienced users to pick up, whereas Azure requires more training and expertise to use. Azure is also more expensive and has extra data transfer costs. According to a comparative study in 2020, users found Azure's documentation hard to read and found that the overall interface was tough to use without in depth training.

(Microsoft, 2022) (Mufti, et al., 2020)

(ProjectPro, 2022)

Google Cloud Platform

Figure 15 – GCP Logo

Costs	Benefits	Limitations
• Provides $300 free credits and free usage of certain products for new users. • Pay only for what service you use	• Cheaper than competitors. • Adjustable pricing. • IaaS, PaaS	• Safety and privacy • Vender pin-down • Very few services compared to competitors

Table 7 – GCP Analysis

Google Cloud Platform (GCP) was launched by Google in 2008. GCP is mainly used by companies for its analytics tools and is not used in the ways that Azure and AWS are used for entire system management. Many companies use GCP in conjunction with Azure or AWS, however, GCP is not compatible with either. GCP is only compatible with other Google products.

GCP does not provide the same functionality as AWS or Azure and has significantly less services on offer. They provide storage, virtual machine capabilities, developer tools, data analysis, and some network monitoring.

One major downside to using GCP is the fact that they monitor any data passed through their services, so any data that a company is analysing using GCP tools is being shared with Google.

(Google, 2022) (Mufti, et al., 2020)

Summary of Cloud Computing Tools

Each of the cloud computing tools mentioned above are extensively used in the computing industry. For my project, I am going to be working exclusively with AWS. Microsoft Azure does provide a certificate management system, but it is not as easily customisable as the AWS ACM is. AWS also allows for its services to be easily integrated with each other through its API, a functionality which will prove extremely useful to me over the course of my project.

Although GCP is a good cloud platform, it simple does not provide any of the functionality I require to carry out my project and design my system for certificate management.

To add to my system, I will also be implementing a resource manager that will keep track of resources that a user is using on AWS. After analysing the vulnerabilities associated with managing resources, I have decided to implement a functionality for checking if any resources have become orphaned, since this is a vulnerability that, similar to SSL certificate expiration, often goes unnoticed.

Since I have decided to use AWS, in the next section of this document I will be examining the AWS tools that will be useful throughout the development of my project.

Overview of Useful AWS Services

Security Hub

Amazon Security Hub is Amazon's solution to security issues within multiple AWS accounts. It integrates with other services such as Amazon Cloud Watch and Amazon Inspector to detect issues within an AWS account and store them in a database within Security Hub. Entries in Security Hub contain: name of issue, AWS resource ID, timestamp, AWS account ID, and other details.

Security Hub picks up some vulnerabilities automatically, but it can be configured manually to pick up other vulnerabilities. In the case of certificate management, Security Hub does not pick up certificates expiring as a vulnerability; if a certificate is expired, but is still stored on ACM, it is still costing the business owner of the account money, so from the point of view of Amazon an expired certificate is an asset.

Once a vulnerability is logged in Security Hub, it is possible to access all the details of the vulnerability through Amazon's API. Findings can also be filtered and exported to a database where they can be manipulated as the user sees fit.

(Amazon Web Services, 2022)

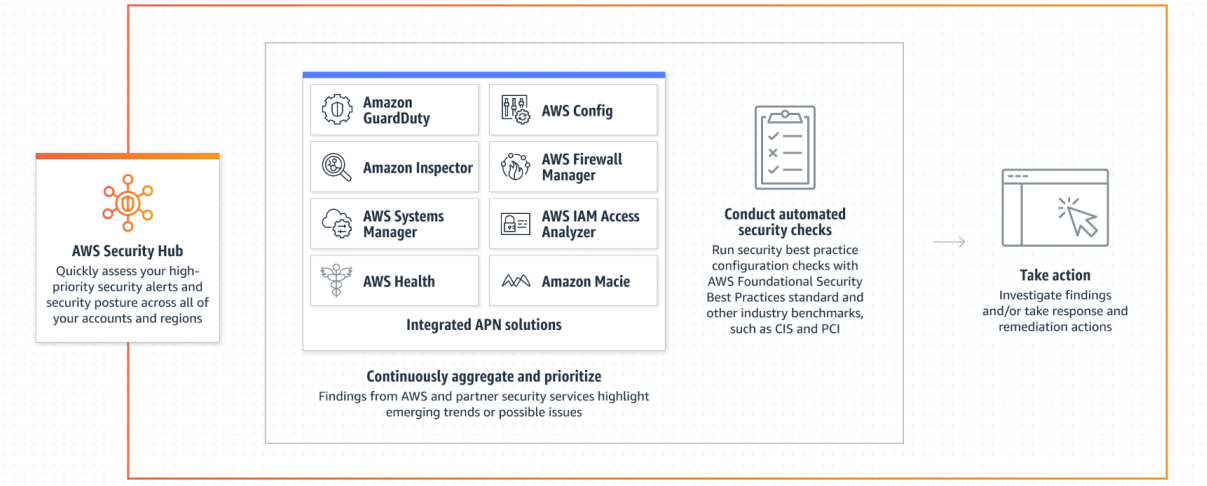

Figure 16 – Security Hub

Lambda & Step Functions

AWS Lambda is a code repository stored on AWS as well as an inbuilt IDE. Program files can be shared or private to an account. Lambda repositories can also be linked to external git repositories and can also be cloned for offline use.

AWS has its own API which allows users to integrate AWS services and resources in their code. AWS also has its own library – boto3 – which contains all the commands used to access these services and resources. Code can be written in many languages including Python, Java, C++, PHP, and many more. Program files can input and output data. For testing, AWS Lambda has a feature to allow users to create test input data and then view what the output is.

Step Functions is a service that allows users to create State Machines. State Machines allow users to integrated multiple program files together from AWS Lambda as well as integrate them with other AWS services. State Machines can then be configured to run on a schedule of the user's choosing.

AWS allows users to write programs in multiple languages, but with Step Functions, each State Machine can have multiple programs all written in different languages since all the State Machine does is pass the output of one program as the input for the next program. As well as this, State Machines have control statements such as if-else statements to allow for branching and loops to allow for recursion.

Step Functions also has a built-in error control system which allows State Machines to have a maximum timeout period, implement timeouts on each individual element of each machine, and configure a retry count on each element.

(Amazon Web Services, 2022)

Simple Notification Service

Amazon Simple Notification Service (SNS) allows users to send notifications via email or text message. To send a notification, users must first collect end point addresses/numbers. These end addresses are then added to an SNS topic. Once they are added they will receive a confirmation message and will have to consent to being added to the SNS topic. Anyone added to an SNS topic will receive the same content, so companies will usually have multiple SNS topics for different teams and employees.

(Amazon Web Services, 2022)

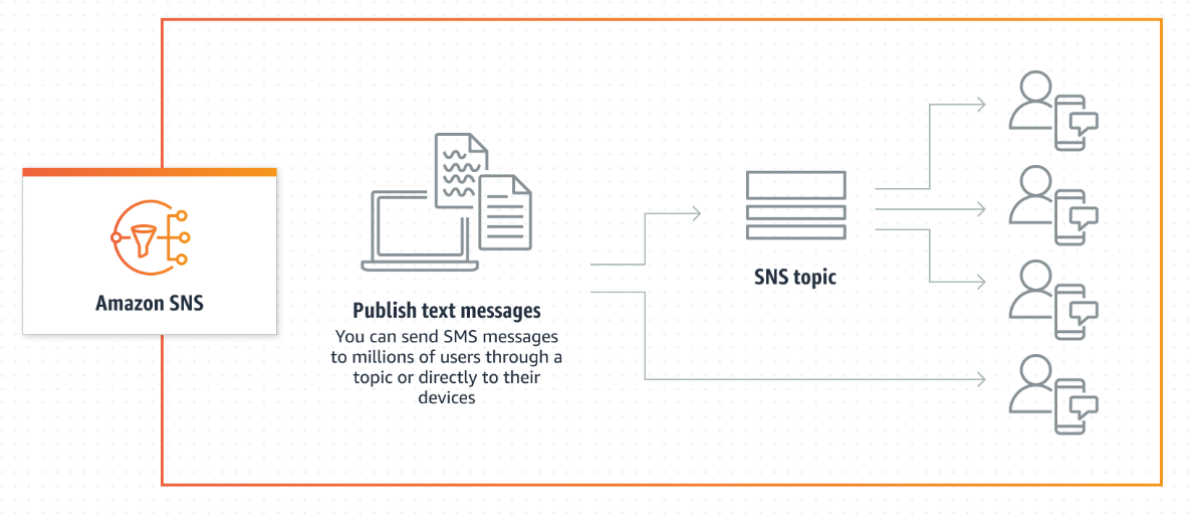

Figure 17 – SNS

Quick Sight Dashboard

Quick Sight Dashboard is a particularly useful AWS service that allows users to display data from within their AWS accounts. Data can be taken from data that is stored within databases or other storage containers within a user's AWS account, as well as being imported form a local file. Quick Sight allows for a lot of customisation in the way that the data can be displayed and allows for multiple charts, diagrams, pages, and dashboards to be created.

(Amazon Web Services, 2022)

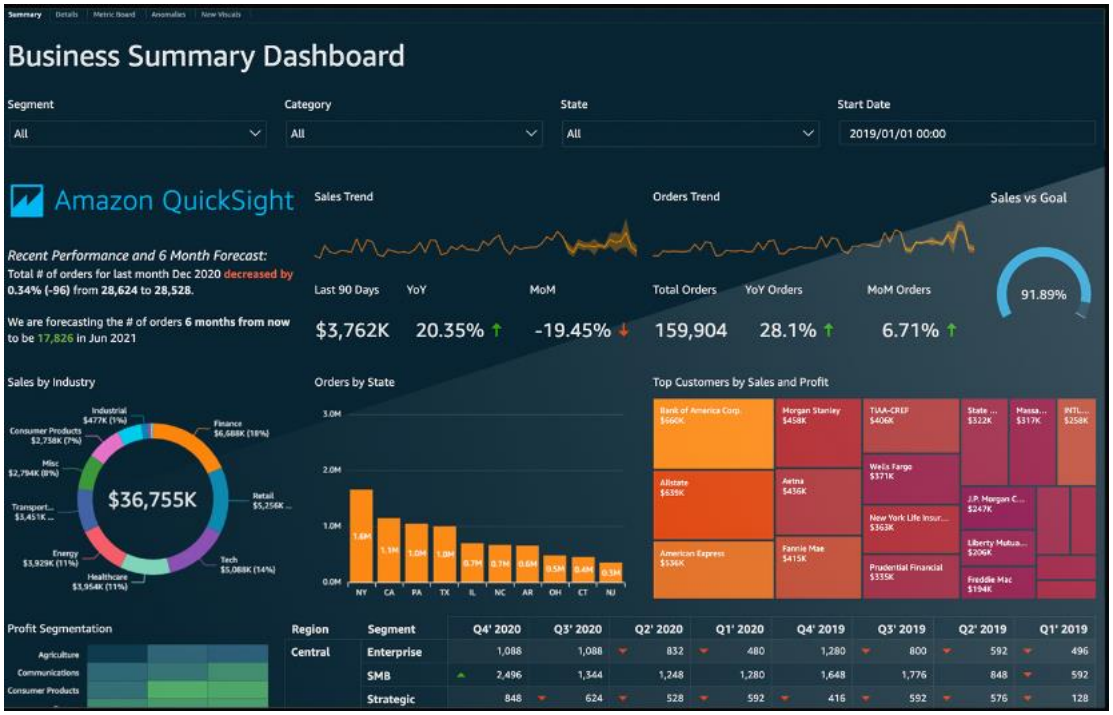

Figure 18 – Sample Dashboard

Programming Languages

AWS allows for programs to be written in multiple languages, and the programming services it provides such as Lambda and Step Functions allow for compatibility between programs written in different languages. Below I will be analysing the pros and cons of using different programming languages and choose which one will be the most beneficial to use during the development of my project.

C++

Figure 19 – C++ Logo

C was created in 1972 and designed by Bell Labs. It is mainly used to for database communication and is also used to design compilers for other programming languages – a compiler is what allows a computer to turn code into instructions. C++ was released by Bell Labs in 1985 as a way to introduce object-oriented programming to C. Object-oriented programming allows for the creation of "objects" which have specific characteristics assigned to them and can also have unique functionality. C++ is very popular for software development, game development, database management, and operating system design.

C++ is currently on version 20 as of December 2020.

Pros	Cons
• Compatible with legacy C code • Uses pointers extensively which allow for dynamic memory allocation • Object-oriented	• Verbose code • Hard to track storage of variables in memory • No garbage collector: unused variables will take up extra space in memory • Hard to read • Hard to implement new libraries

Table 8 – C++ Pros & Cons

(Data Flair, 2022) (Bhave & Patekar, 2012)

(Manley, 2020) (Duggal, 2022)

Java

Figure 20 – Java Logo

Java is a popular programming language that was released by Sun Microsystems in 1995. It stems from C and C++, so the structure is very similar. Like C++, Java is an object-oriented language. Java is a popular language for database management, game design, system administration, and software development.

Java is currently on version 19 as of September 2022.

Pros	Cons
• Easier to read than C++ • Object-oriented • Allows for inheritance: characteristics can be passed down between similar objects • Good community support • Automatic garbage collection: unused variables are deleted from memory to free up space • Regularly updated	• Verbose code • Can be difficult to read • More memory required • Slow processing

Table 9 – Java Pros & Cons

(Data Flair, 2022) (Oracle, 2022)

(Behler, 2022)

Python

Figure 21 – Python Logo

Python was first released in 1991 by Guido van Rossum and was named after Monty Python's comedy series "Monty Python's Flying Circus". Python has grown in popularity over the past 20 years and is one of the most popular languages for new developers thanks to its simple and easy to learn design. Python is highly versatile, and can be used in software development, robotic automation, Graphical User Interface (GUI) design, machine learning and Artificial Intelligence (AI), and game design.

Python is currently on version 3.11.0 as of October 2022.

Pros	Cons
• Easy to learn • Easy to read • Less verbose than other languages • Easy to implement new libraries • Highly scalable • Regularly updated	• Slow processing • High memory consumption

Table 10 – Python Pros & Cons

(Bhagat, 2022) (Python Institute, 2022)

(Python, 2022)

Perl

Figure 22 – Perl Logo

Perl was first developed in 1987 by American programmer Larry Wall. Perl is not exceedingly popular among software developers, but AWS allows developers to write their code in Perl if they wish to do so. Perl can be useful for database management and network administration; however, it is not recommended to use Perl for creating interfaces or for web development.

Perl is currently on version 5.36.0 as of

Pros	Cons
• Open source • Can easily manipulate text • Supports external libraries from other languages such as C and C++	• Difficult to fix bugs • Not widely used so not much community support

Table 11 – Perl Pros & Cons

(ProsCons, 2020) (Perl, 2022)

Summary of Programming Languages

AWS designed its infrastructure in a way that a user could create an application while utilising and integrating different programming languages throughout the development of their application. Java is a viable option that could provide the I want to bring about with this project, but for my system, I have decided to use Python for a number of reasons:

- Python allows users to easily implement new libraries, which will come in useful when implementing the AWS Boto3 library during production of my system.
- Python is easy to read and is not as verbose as other programming languages, and one of the key aspects of my project is that I want my system to be designed to be adaptable and scalable; I want to make it as easy as possible for a security team who could be using my system to adapt my system to meet their needs and this will be a lot easier if they can read and understand my code.
- Python is designed in a way that makes it easier for a developer to learn new commands quickly and efficiently, which I will need to do a lot throughout my project.

Although the majority of my project will be written in Python, because if the way AWS is designed, I can choose to write a function or an entire program in another language if I need to over the course of my project.

Summary & Conclusion

After researching into certificate management and analysing multiple certificate management tools, I have decided that my aim with this project will be to make the tracking of certificates within AWS easier by creating a logging system which will be connected to a notification system as well as a dashboard. This system will be designed entirely within AWS using native AWS tools such as AWS Lambda, AWS Step Functions, AWS SNS, and AWS Quick Sight Dashboard.

I chose to design my tool within AWS because businesses are rapidly expanding in the cloud, so designing a certificate management platform native to a cloud infrastructure would be greatly beneficial. Although AWS already has a certificate management platform, from my experience working in the identity and access management field, I'm aware that currently certificate management on the platform is a very manual process. By automating this process, I aim to cut down on the time an employee would have to spend monitoring on managing certificates, remove the element of human error during the management and renewal of certificates, and cut down on the cost a company that is using AWS would have to spend on monitoring their certificates.

The programming language I've chosen to use for my project is Python for a number of reasons: it's regularly updated, less verbose than other languages so it will be easier for other programmers to read my code, and its design will make it easier to pick up the AWS Boto3 library than with another language.

Although certificate management is the primary focus of this project, I plan on creating a system that will also serve to act as a template for a more extensive monitoring system that could be adapted to monitor different vulnerabilities. To do this I will need to design my system with scalability and modularity in mind, meaning that I would ideally like the solutions I'll be creating to be designed in a way that they can be reused and expanded upon with ease.

Glossary

MITM – Man in the Middle

SSL – Secure Sockets Layer

PCT – Private Communication Technology

IETF – Internet Engineering Task Force

TLS – Transport Layer Security

TLSWG – Transport Layer Security Working Group

CA – Certificate Authority

RSA – Rivest–Shamir–Adleman

ECC – Elliptic Curve Cryptography

IAM – Identity & Access Management

API – Application Programming Interface

SSH – Secure Shell

AD – Active Directory

AWS – Amazon Web Services

ACM – Amazon Certificate Manager

SAM – Server & Application Monitor

IaaS – Infrastructure as a Service

PaaS – Platform as a Service

SaaS – Software as a Service

GCP – Google Cloud Platform

SNS – Simple Notification Service

GUI – Graphical User Interface

AI – Artificial Intelligence

Bibliography

Alam, S., Muqeem, M. & Khan, S. A., 2018. Review on Security Aspects for Cloud Architecture. *International Journal of Electrical and Computer Engineering*, 8(5), pp. 3129-3139.

Amazon Web Services, 2022. *Amazon Certificate Manage.* [Online]
Available at: https://www.amazonaws.cn/en/certificate-manager/
[Accessed 31 08 2022].

Amazon Web Services, 2022. *Amazon QuickSight.* [Online]
Available at: https://aws.amazon.com/quicksight/
[Accessed 15 11 2022].

Amazon Web Services, 2022. *Amazon SNS.* [Online]
Available at: https://aws.amazon.com/sns/
[Accessed 21 10 2022].

Amazon Web Services, 2022. *AWS Lambda.* [Online]
Available at: https://aws.amazon.com/lambda/
[Accessed 21 10 2022].

Amazon Web Services, 2022. *AWS Step Functions.* [Online]
Available at: https://aws.amazon.com/step-functions/
[Accessed 21 10 2022].

Amazon Web Services, 2022. *What is AWS Security Hub?.* [Online]
Available at: https://docs.aws.amazon.com/securityhub/latest/userguide/what-is-securityhub.html
[Accessed 21 10 2022].

Amazon, 2022. *AWS.* [Online]
Available at: https://aws.amazon.com/
[Accessed 15 11 2022].

Behler, M., 2022. *Java Versions and Features.* [Online]
Available at: https://www.marcobehler.com/guides/a-guide-to-java-versions-and-features#:~:text=What%20is%20the%20latest%20Java,17%2C%20released%20in%20September%202021.
[Accessed 22 11 2022].

Bhagat, V., 2022. *Pros and Cons of Python Programming Language.* [Online]
Available at: https://www.pixelcrayons.com/blog/python-pros-and-cons/
[Accessed 22 11 2022].

Bhan, R. et al., 2019. *VM Availability in Presence of Malicious Attacks in Open-Source Cloud.* Noida, India, IEEE.

Bhardwaj, R., 2020. *Public vs Private Certificate Authority.* [Online]
Available at: https://ipwithease.com/public-vs-private-certificate-authority/
[Accessed 18 11 2022].

Bhave, M. & Patekar, S., 2012. Brief History of C++. In: *Object Oriented Programming with C++*. s.l.:Pearson India.

Cloudflare, 2022. *What is a TLS handshake?.* [Online]
Available at: https://www.cloudflare.com/en-gb/learning/ssl/what-happens-in-a-tls-handshake/
[Accessed 15 11 2022].

Dabrowsk, C. & Mills, K., 2011. *VM Leakage and Orphan Control in Open-Source Clouds.* Athens, Greece, IEEE.

Data Flair, 2022. *Advantages and Disadvantages of C++*. [Online]
Available at: https://data-flair.training/blogs/advantages-and-disadvantages-of-cpp/
[Accessed 22 11 2022].

Data Flair, 2022. *Pros and Cons of java.* [Online]
Available at: https://data-flair.training/blogs/pros-and-cons-of-java/
[Accessed 22 11 2022].

DigiCert, 2022. *CertCentral TLS/SSL Manager.* [Online]
Available at: https://www.digicert.com/tls-ssl/certcentral-tls-ssl-manager#features
[Accessed 31 08 2022].

DigiCert, 2022. *How Does SSL/TLS Work?.* [Online]
Available at: https://www.websecurity.digicert.com/security-topics/how-does-ssl-handshak
[Accessed 18 08 2022].

DigiCert, 2022. *What is SSL, TLS and HTTPS?.* [Online]
Available at: https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https
[Accessed 17 October 2022].

Duggal, N., 2022. *Top 12 Uses of C++*. [Online]
Available at: https://www.simplilearn.com/tutorials/cpp-tutorial/top-uses-of-c-plus-plus-programming#:~:text=in%20the%20Field-,What%20is%20C%2B%2B%20Used%20For%3F,engineering%2C%20data%20structures%2C%20etc.
[Accessed 22 11 2022].

Educative, 2022. *What is the RSA algorithm?.* [Online]
Available at: https://www.educative.io/answers/what-is-the-rsa-algorithm
[Accessed 13 11 2022].

Foxe, K., 2022. *Over 200 calls to Garda unanswered as fallback emergency system also fails during 999 outage.* [Online]
Available at: https://www.irishexaminer.com/news/arid-40967141.html
[Accessed 17 03 2023].

GeeksForGeeks, 2022. *RSA Algorithm in Cryptography.* [Online]
Available at: https://www.geeksforgeeks.org/rsa-algorithm-cryptography/
[Accessed 13 11 2022].

Google, 2022. *Google Cloud.* [Online]
Available at: https://cloud.google.com/
[Accessed 15 11 2022].

Grubbs, P., 2022. *Public vs Private Certificate Authority.* [Online]
Available at: https://www.securew2.com/blog/public-vs-private-certificate-authority#:~:text=Where%20a%20public%20CA%20hands,cyclical%20or%20time%2Dsensitive%20nature.
[Accessed 18 11 2022].

IBM Cloud Education, 2021. *IaaS vs PaaS vs SaaS.* [Online]
Available at: https://www.ibm.com/cloud/learn/iaas-paas-saas#toc-paas-zUewploH
[Accessed 17 11 2022].

IBM, 2021. *Public certificates vs private certificates.* [Online]
Available at: https://www.ibm.com/docs/en/i/7.2?topic=dcm-public-certificates-versus-private-certificates
[Accessed 18 11 2022].

Islam, T., Manivannan, D. & Zeadally, S., 2016. A Classification and Characterization of Security Threats in Cloud Computing. *International Journal of Next-Generation Computing,* 7(1), pp. 268-285.

Kapoor, V., Abraham, V. S. & Singh, R., 2008. Elliptic Curve Cryptography. *ACM Ubiquity,* May.9(20).

KeyFactor, 2022. *What is a Certificate Authority? Everything Need to Know About Public & Private CAs.* [Online]
Available at: https://www.keyfactor.com/resources/what-is-a-certificate-authority/
[Accessed 18 11 2022].

Kute, V., Paradhi, P. & Bamnote, G., 2009. A SOFTWARE COMPARISON OF RSA AND ECC. *International Journal Of Computer Science And Applications,* 2(1).

Mahajan, D. P. & Sachdeva, A., 2013. A Study of Encryption Algorithms AES, DES and RSA for Security. *Global Journal of Computer Science and Technology Network, Web & Security,* 12(15).

Mahto, D. & Yadav, D. K., 2017. RSA and ECC: A Comparative Analysis. *International Journal of Applied Engineering Research,* 12(19).

ManageEngine, 2022. *Key Manager Plus.* [Online]
Available at: https://www.manageengine.com/key-manager/
[Accessed 31 08 2022].

Manley, G., 2020. *A Brief History of C Programming.* [Online]
Available at: https://www.section.io/engineering-education/history-of-c-programming-language/#:~:text=The%20C%20programming%20language%20came,the%20nascent%20Unix%20operating%20system.
[Accessed 22 11 2022].

Microsoft, 2022. *Azure.* [Online]
Available at: https://azure.microsoft.com/en-us/
[Accessed 15 11 2022].

Mufti, T., Mittal, P. & Gupta, B., 2020. *A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services.* New Delhi, Department of Computer, Science and Engineering, SEST.

Munir, K. & Palaniappan, S., 2013. Secure Cloud Architecture. *Advanced Computing: An International Journal,* 4(1).

Oracle, 2022. *What is Java technology and why do I need it?.* [Online]
Available at: https://www.java.com/en/download/help/whatis_java.html
[Accessed 22 11 2022].

Perl, 2022. *About Perl.* [Online]
Available at: https://www.perl.org/about.html
[Accessed 22 11 2022].

Prodromou, A., 2019. *TLS Security 2: A Brief History of SSL/TLS.* [Online]
Available at: https://www.acunetix.com/blog/articles/history-of-tls-ssl-part-2/#:~:text=The%20Secure%20Sockets%20Layer%20(SSL,it%20had%20serious%20security%20flaws.
[Accessed 11 11 2022].

ProjectPro, 2022. *AWS vs Azure-Who is the big winner in the cloud war?.* [Online]
Available at: https://www.projectpro.io/article/aws-vs-azure-who-is-the-big-winner-in-the-cloud-war/401
[Accessed 17 11 2022].

ProsCons, 2020. *Pros and cons of Perl Programming Language.* [Online]
Available at: https://pros-cons.net/pros-and-cons-of-perl-programming-language/
[Accessed 22 11 2022].

Python Institute, 2022. *Python - the language of today and tomorrow.* [Online]
Available at: https://pythoninstitute.org/about-python#:~:text=Python%20was%20created%20by%20Guido,called%20Monty%20Python's%20Flying%20Circus.
[Accessed 22 11 2022].

Python, 2022. *Python Documentation by Version.* [Online]
Available at: https://www.python.org/doc/versions/#:~:text=Python%203.11.,released%20on%2024%20October%202022.
[Accessed 22 11 2022].

SolarWinds, 2022. *SSL Certificate Management and Expiration Monitoring Tool.* [Online]
Available at: https://www.solarwinds.com/server-application-monitor/use-cases/ssl-certificate-monitor
[Accessed 31 08 2022].

TechTarget, 2022. *What is AWS?.* [Online]
Available at: https://www.techtarget.com/searchaws/definition/Amazon-Web-Services
[Accessed 15 11 2022].

Thomas, S., 2000. *SSL and TLS Essentials.* s.l.:John Wiley & Sons, Inc.

Zhou, X. & Tang, X., 2011. *Research and implementation of RSA algorithm for encryption and decryption.* s.l., IEEE.