FMISD13207 Cloud Security Technologies

Kęstutis Daugėla

2022-05-08

Contents

Introduction

Topics from FMISD13207 Cloud Security Technologies module:

Information Security Fundamentals (information security problematics, classification and evolution of threats, identification, authentication, access control, security principals, strategies, models, taxonomies and antologies); **Cryptography (simetric and public key cryptography**, DES, AES, **RSA**. stream ciphers, **cryptographic protocols, authentication, electronic signature, management of electronic identity**); **Network Security (routing, firewalls**, VPN, **web security**, **network**

perimeter protection, host-level protection, **authentication technologies**); **Attacking Information Technology Systems (attack types, real-life case studies**, intrusion detection, formal analysis techniques); **Information Security Technologies** (antivirus, IDS, **host and perimeter protection systems**, Honeypots); **Implementing Effective Information Security Programs** (legal, regulatory and privacy issues, **security standards, security best practices, security policy**).

0.1 Introduction to Cloud

The underlying concept of cloud computing was introduced way back in 1960s by John McCarthy in his book, "The challenge of the Computer Utility". His opinion was that "computation may someday be organized as a public utility." The rest became history and the majority of the software used now is running in the cloud seamlessly (Surbiryala and Rong, 2019).

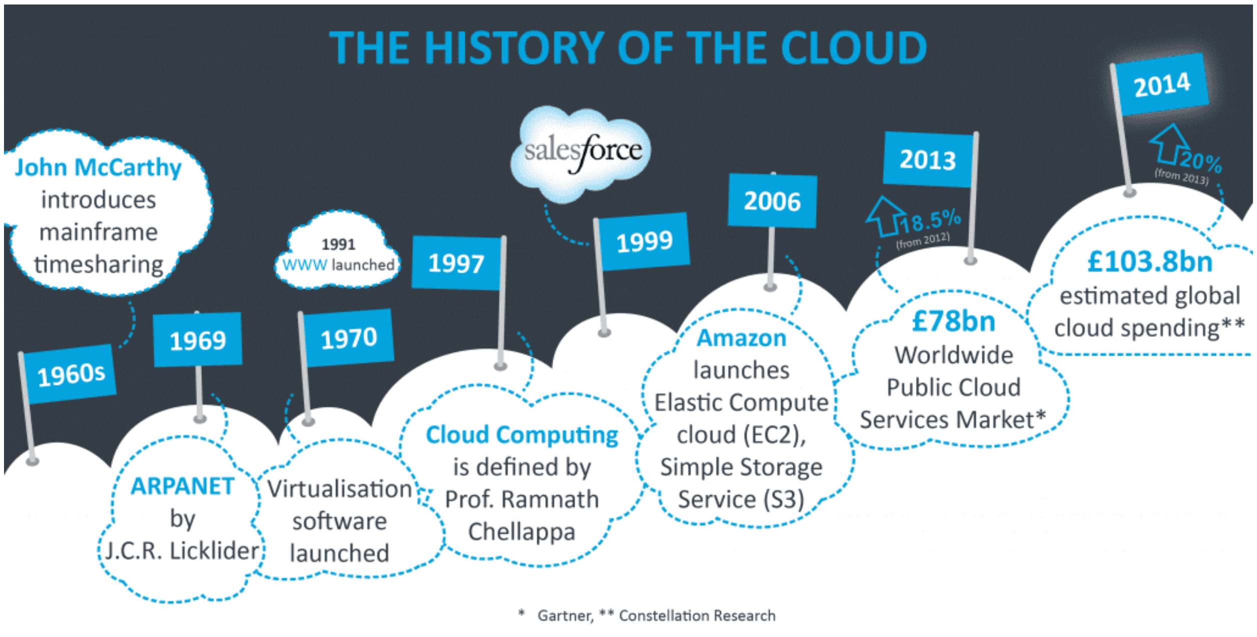

THE HISTORY OF THE CLOUD
John McCarthy introduces mainframe timesharing
1960s
1969
ARPANET by J.C.R. Licklider
1970
Virtualisation software launched
1991 WWW launched
1997
Cloud Computing is defined by Prof. Ramnath Chellappa
salesforce
1999
2006
Amazon launches Elastic Compute cloud (EC2), Simple Storage Service (S3)
2013
18.5% (from 2012)
£78bn Worldwide Public Cloud Services Market*
2014
20% (from 2013)
£103.8bn estimated global cloud spending**
* Gartner, ** Constellation Research

Figure 1: The history of the cloud - image source https://itchronicles.com/

Cloud can solve a lot of problems nowadays - starting with reduced cost, enhanced security, and flexible approach (Srivastava and Khan, 2018) up to sustainability (Parthasarathy and Kumar, 2012) and accessibility around the world. Continuous Integration and Deployment (CI/CD) is easier than even treating now only the applications, but the whole infrastructure as code. This leads to enhanced productivity and cost optimization (Garg and Garg, 2019).

Is there anything revolutionary in the cloud offerings today? Definitely, no - people used these capabilities for ages. The only difference is the scale and popularity these days.

Cloud services usually are grouped into three categories:

- SaaS (Software as a service) is a software distribution model in which a cloud provider hosts applications and makes them available to end-users over the internet
- PaaS (Platform as a service) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications
- IaaS (Infrastructure as a service) is a type of cloud computing service that offers essential compute, storage, and networking resources on-demand, on a pay-as-you-go basis

However, despite the gain achieved from cloud computing, organizations are slow in fully accepting it due to security issues and challenges associated with it (Bairagi and Bang, 2015). However, almost every cloud-

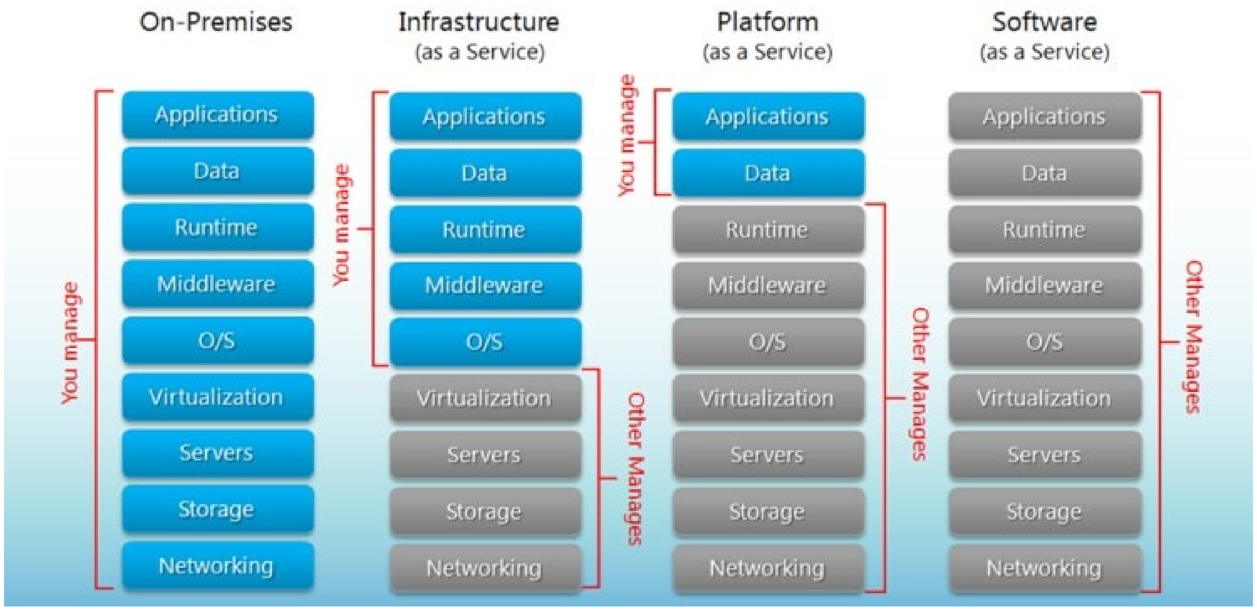


Figure 2: IaaS vs PaaS vs SaaS - image source https://www.bigcommerce.com/blog/saas-vs-paas-vs-iaas/

ready company uses the public cloud (97%) to some extent leaving hybrid cloud setup the dominant one (78%). Companies rarely use public or private cloud alone (19% vs 2% respectively). According to Forbes, there are now 77 % of organizations, having one or some parts of their systems in the cloud.

0.2 Introduction to Cloud Security

Cloud service providers follow a shared security responsibility model, which means that your security team retains some security duties as you migrate applications, data, containers, and workloads to the cloud, while the provider takes part, but not all, of the responsibility. Clearly defining your duties from those of your providers is critical for minimizing the risk of introducing vulnerabilities into your public, hybrid, or multi-cloud systems, as shown in the graph below.

Cloud challenges and directions have been categorized into 6 groups: Security, Autonomic Resource Management (ARM), Cloud Adoption, Cloud development and benchmarking big data technologies and cloud computing and social clouds. Security is the most common challenge of cloud computing as the paper indicates (which is the outcome of literature review of over 190 resources including books, journal and conference papers, research project reports and deliverables, European Commission roadmaps and calls for proposals, online weblogs, white papers, business reviews, company websites, profiles and offered services by cloud providers). (Keshavarzi et al., 2020)

One of the main findings of other study was that the most targeted type of organisation are medical organisations and Business Services Organisations since they possess the most sensitive personal data. Moreover, it was observed that in the recent years the frequency hacking breaches on the two sectors has intensified, with an inter-arrival time of 4 and 7.5 days respectively. These findings implies taking serious actions to secure personal data especially for MED and BSO organization (Hammouchi et al., 2019). Another public data breach dataset of the biggest hacks confirms these findings.

Sector	Attacks	Number of records lost
web	108	7009835665
government	44	922822573
health	43	174669929

Sector	Attacks	Number of records lost
finance	39	1162050100
retail	35	882555730
tech	23	1590382232
telecoms	22	837584000
app	21	553259000
transport	16	181626000
gaming	15	224490755
academic	11	5953000
misc	10	90780000
military	5	82227432

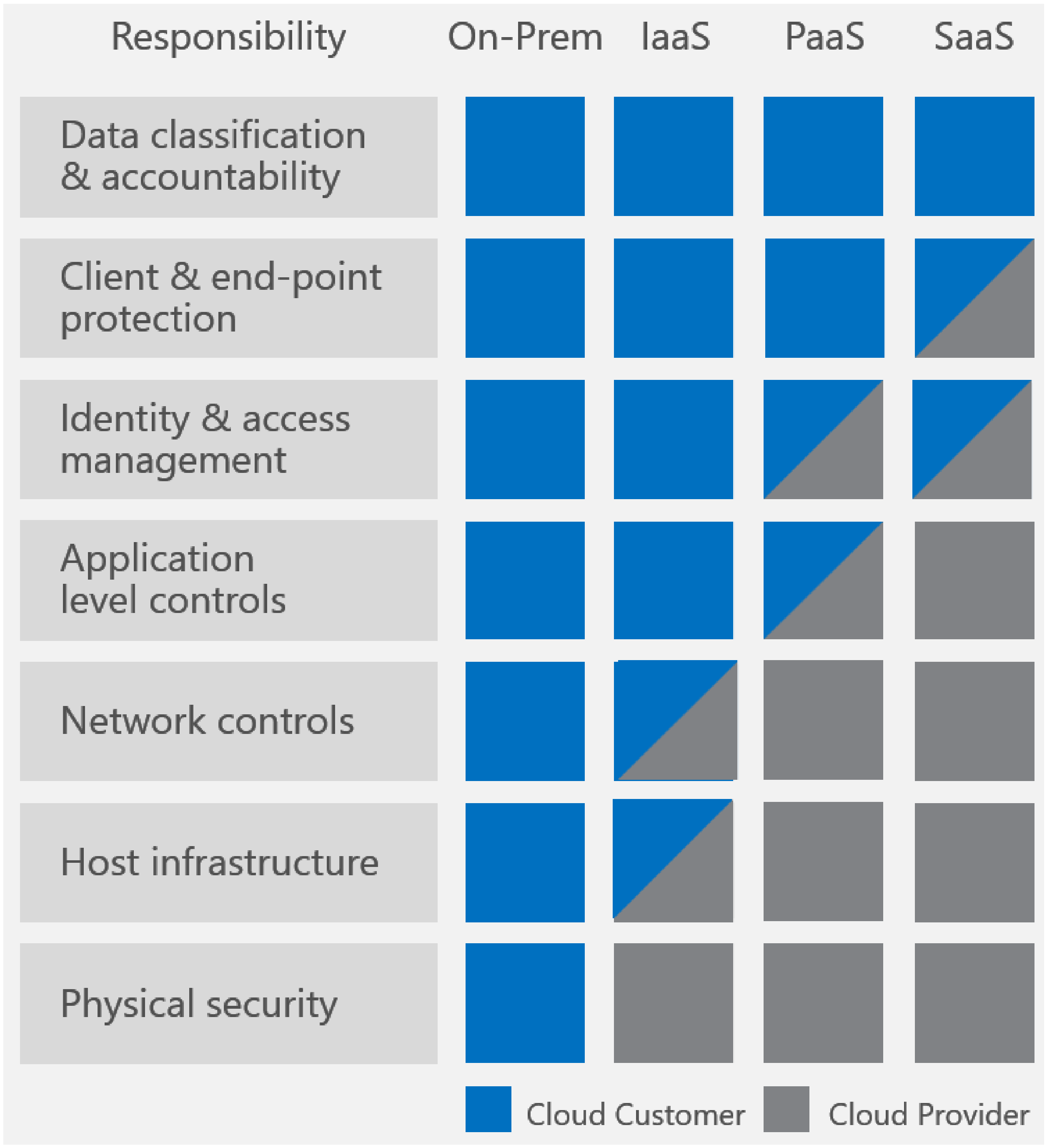
Even such tech companies as Facebook, Twitter, Amazon, Twitch were hacked successfully making a considerable damage on their reputation and their clients.

The method's classification is not very precise and very overlapping; for example, what is termed a "oops!" could be classified as bad security or vice versa. On the other hand, hacking methods are not described, and hence we shall explore attack types in greater detail in the second chapter.

1 Type of attacks

The literature has examined a variety of security flaws(Humayun et al., 2020). To aid in comprehending some of the most prevalent cyber security weaknesses, the following are detailed:

- **Malware**. For the previous decade, malware attacks have been the most serious cyber security danger to many enterprises (Lim and Lukito, 2013). Malware The attacker uses malicious software to gain unauthorized access to computer systems by exploiting their security flaws. Malware is motivated by an extreme financial or political gain, which increases an attacker's drive to compromise as many network devices as possible in order to accomplish their harmful goals. Viruses, worms, trojans, backdoors and adware are but a few examples that fall under the umbrella of malware (Mokoena and Zuva, 2017).
- **DDoS**. Cyber security is comprised on three essential components: confidentiality, integrity, and availability. Denial of Service (DoS) attacks and their version, Distributed Denial of Service (DDoS), are conceivable threats that deplete system resources, rendering them inaccessible to authorized users, hence breaching one of the security components—availability. DoS attacks on networks are widespread and have the potential to be catastrophic. Numerous types of DoS attacks have been identified thus far, and the most of them are extremely efficient at disrupting network connectivity. Both IPv4 and IPv6 are quite vulnerable to these attacks (Tripathi and Mehtre, 2013). The frequency and scale of attacks have escalated in recent years, from a few megabytes to hundreds of gigabytes. It is difficult to identify these attacks efficiently due to changes in attack patterns or new forms of attacks (Vanitha et al., 2017).
- **Phishing** is a very effective approach of cybercrime that criminals use to fool consumers and steal critical data. Since the first phishing assault was disclosed in 1990, the attack vector has grown into a more sophisticated attack vector. At the moment, phishing is regarded to be one of the most prevalent forms of fraud on the Internet. Phishing attacks can result in significant losses for their victims, including sensitive data, identity theft, businesses, and state secrets. (Alkhalil et al., 2021)
- **SQL Injection Attack (SQLIA)** is one of the most terrifying risks to web applications. Input validation flaws were the cause of a SQL injection attack on the web. SQLIA is a harmful behavior that exploits invalid SQL statements in order to exploit data-driven applications. This vulnerability allows an attacker to exploit manipulated input to gain access to the application's back-end databases through the application's interaction with them. As a result, the attacker can acquire access to the database without obtaining genuine clearance by introducing, changing, or removing key information (Hlaing and Khaing, 2020).



Responsibility	On-Prem	IaaS	PaaS	SaaS
Data classification & accountability	🟦	🟦	🟦	🟦
Client & end-point protection	🟦	🟦	🟦	🟦/⬛
Identity & access management	🟦	🟦	🟦/⬛	🟦/⬛
Application level controls	🟦	🟦	🟦/⬛	⬛
Network controls	🟦	🟦/⬛	⬛	⬛
Host infrastructure	🟦	🟦/⬛	⬛	⬛
Physical security	🟦	⬛	⬛	⬛

🟦 Cloud Customer ⬛ Cloud Provider

Figure 3: Shared responsibility in the cloud - image source www.microsoft.com

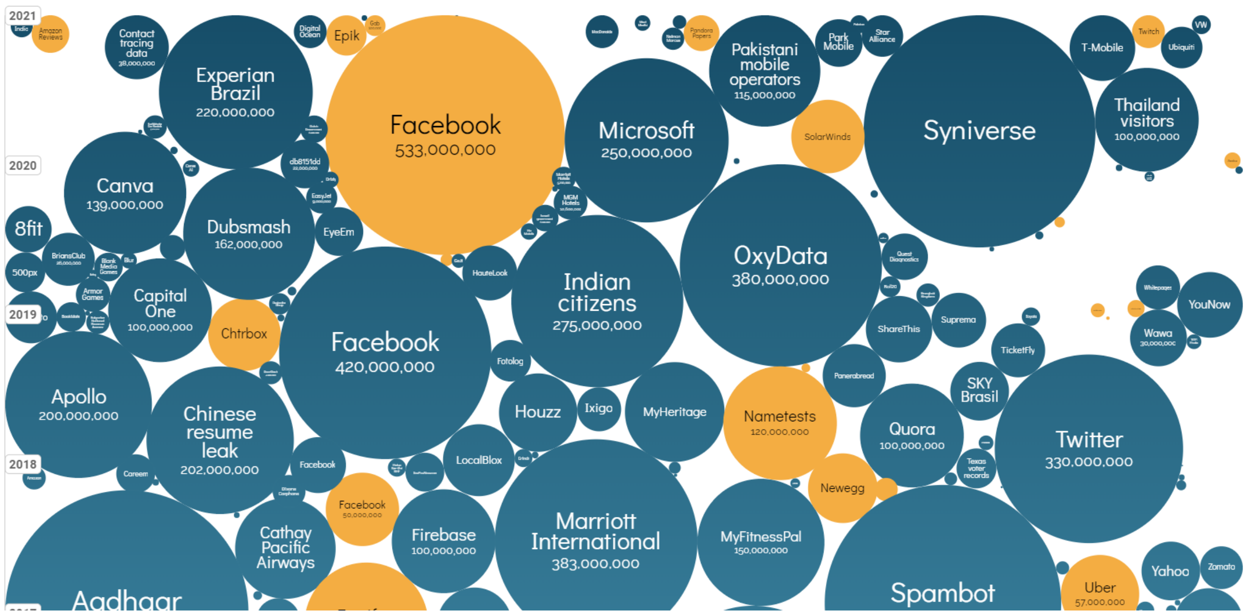


Figure 4: World's Biggest Data Breaches & Hacks - image source https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

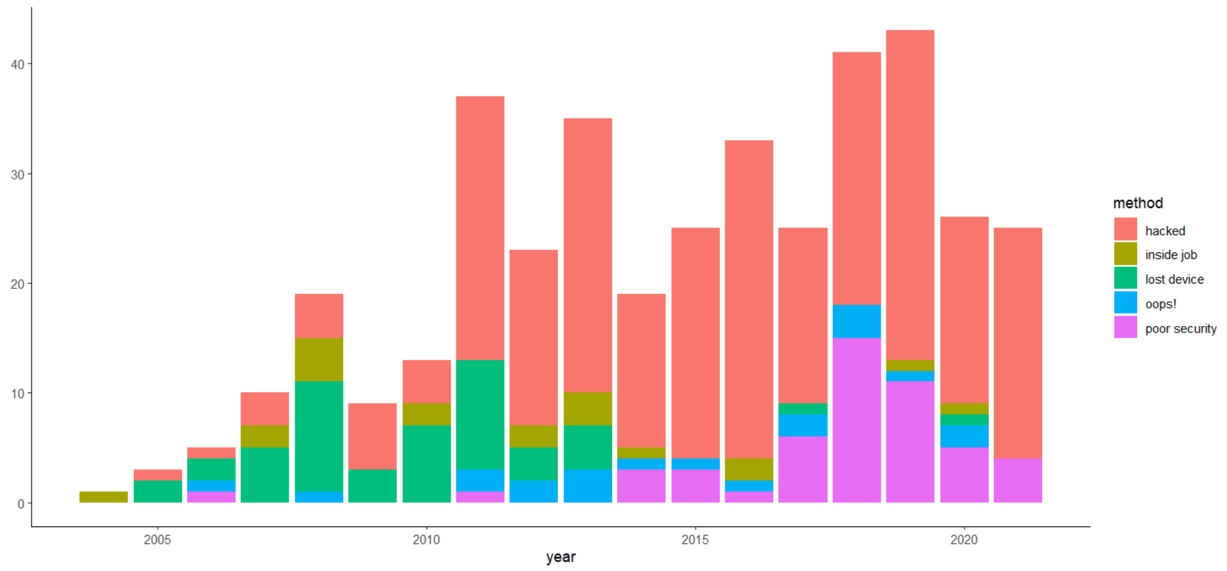


Figure 5: Hacks sorted by method

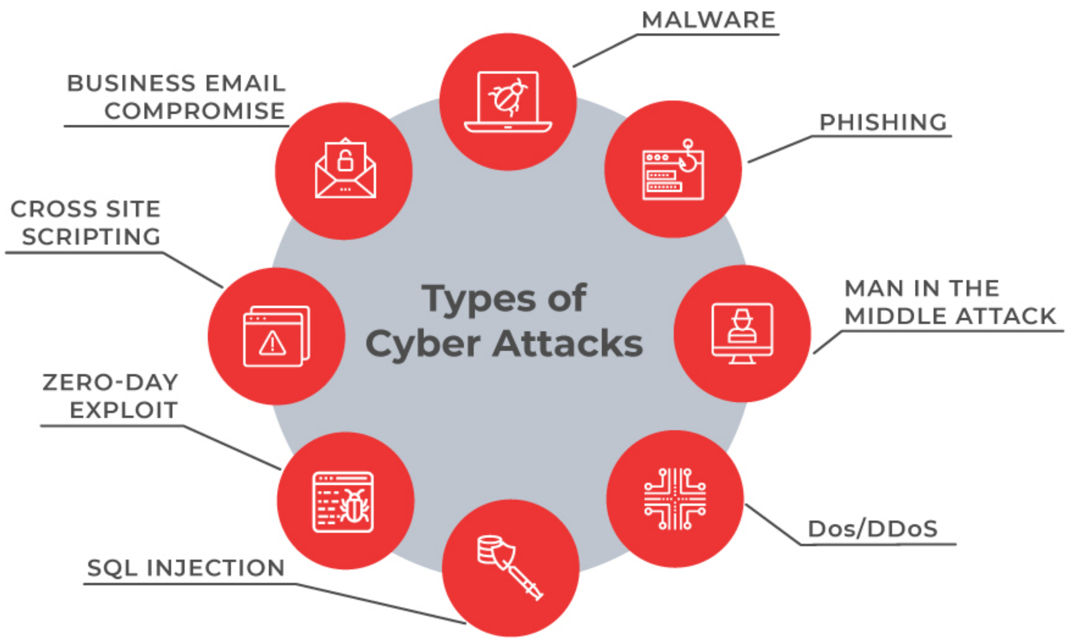


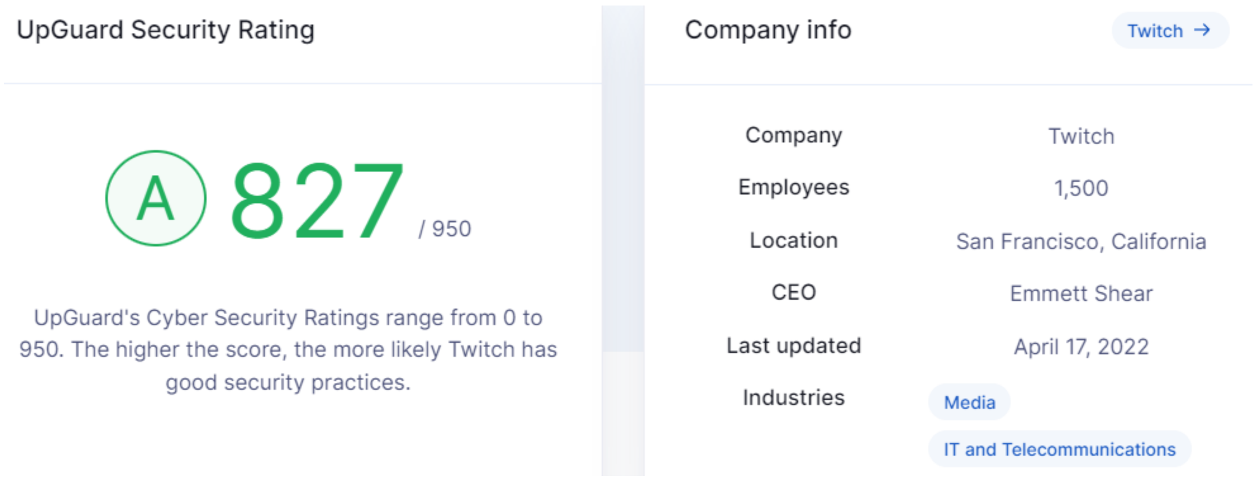
Figure 6: The Vulnerabilities that a Business can Experience - image source https://blog.ecosystm360.com/cyber-attacks-threats-risks/

- **Man-in-the-middle**. An attack in which an outsider or third party infiltrates the space between two online users while both users are unaware. In this instance, the malware primarily monitors and has the capacity to modify the information classified exclusively to these two people. Generally, it is referred to as a protocol to refer to an unauthorized user within the system who has the ability to view and modify the system's data without leaving a trail for the system's existing users (Javeed and MohammedBadamasi, 2020).
- **Cross-site scripting** is a serious issue in Web Applications. With more connected devices that use a variety of Web Applications for various tasks, the potential of XSS assaults grows. By exploiting XSS vulnerabilities in Web applications, hackers can steal victims' session details or other sensitive information(Nagarjun and Ahamad, 2020).
- **Zero-day exploit** continue to be a significant security concern to enterprises. When a vendor becomes aware of a zero-day vulnerability, releasing a fix in a timely manner becomes a priority due to the possibility of zero-day exploits. However, we continue to lack knowledge on the factors that influence the time it takes for such vulnerabilities to be patched. It was discovered that while IT companies are quick to release timely updates for zero-day vulnerabilities that affect multiple vendors, products, and versions, vulnerabilities that require privileges and compromise confidentiality are less likely to be patched on time. (Roumani, 2021).
- **BEC (Business Email Compromise)** is a sophisticated email fraud scheme that targets firms who deal with international suppliers and frequently transfer payments via wire transfers. BEC attacks are designed to eliminate security protections by capitalizing on flaws in human behavior and decision-making (Agazzi, 2020).

2 Data Breach - Case Study

2.1 Twitch

Twitch is a global community that comes together each day to create multiplayer entertainment: unique, live, unpredictable experiences created by the shared interactions of millions. Acquired by Amazon in 2014.


UpGuard Security Rating

A 827 / 950

UpGuard's Cyber Security Ratings range from 0 to 950. The higher the score, the more likely Twitch has good security practices.

Company info

Company	Twitch
Employees	1,500
Location	San Francisco, California
CEO	Emmett Shear
Last updated	April 17, 2022
Industries	Media, IT and Telecommunications

Figure 7: UpGuard Security Rating - image source https://www.upguard.com/

Twitch recently suffered a data breach that, according to security analysts, may have revealed extensive information about the platform's computer code, security vulnerabilities, and payments to content providers. According to the source, the file contained the history of Twitch's source code; proprietary software development kits; an unreleased competitor to Steam, an online games store; programs used by Twitch to test its own security vulnerabilities; and a list of the amount of money earned by each of the site's streamers since 2019 NY Times - A 'potentially disastrous' data breach hits Twitch, the livestreaming site

As another security proffessional indicates in medium.com - Thoughts on the Twitch Breach: the attackers did not gain access through a zero-day vulnerability or a supply chain assault. Something existed in a state that was not compliant with or anticipated of it.

It appears that the problem was a misconfiguration as this statement says something was in error. The attackers did not get in due to a zero-day or supply chain attack. Something existed in a non-compliant or expected state. It sounds as though a server was running something on an Internet-facing port. However, the exposed data could have existed in an S3 bucket and all we know is that a server entered an undesirable state, which resulted in data disclosure. The misconfiguration could be as simple as a server being allowed excessive permissions and access, or as complex as a server being exposed to the Internet when it should have been in a private network.

Many people host their data on AWS and might be thinking, "If Amazon can't keep data secure on AWS, who can?".

Assess that you have zero-trust networking and permissions that provide persons and applications only the access they require. In an enterprise of that size, source control systems, if that is what was impacted in this case, should never be exposed directly to the entire Internet. Small enterprises may do so temporarily to stay afloat, but the majority of source control solutions allow you to restrict access to specific IP ranges.

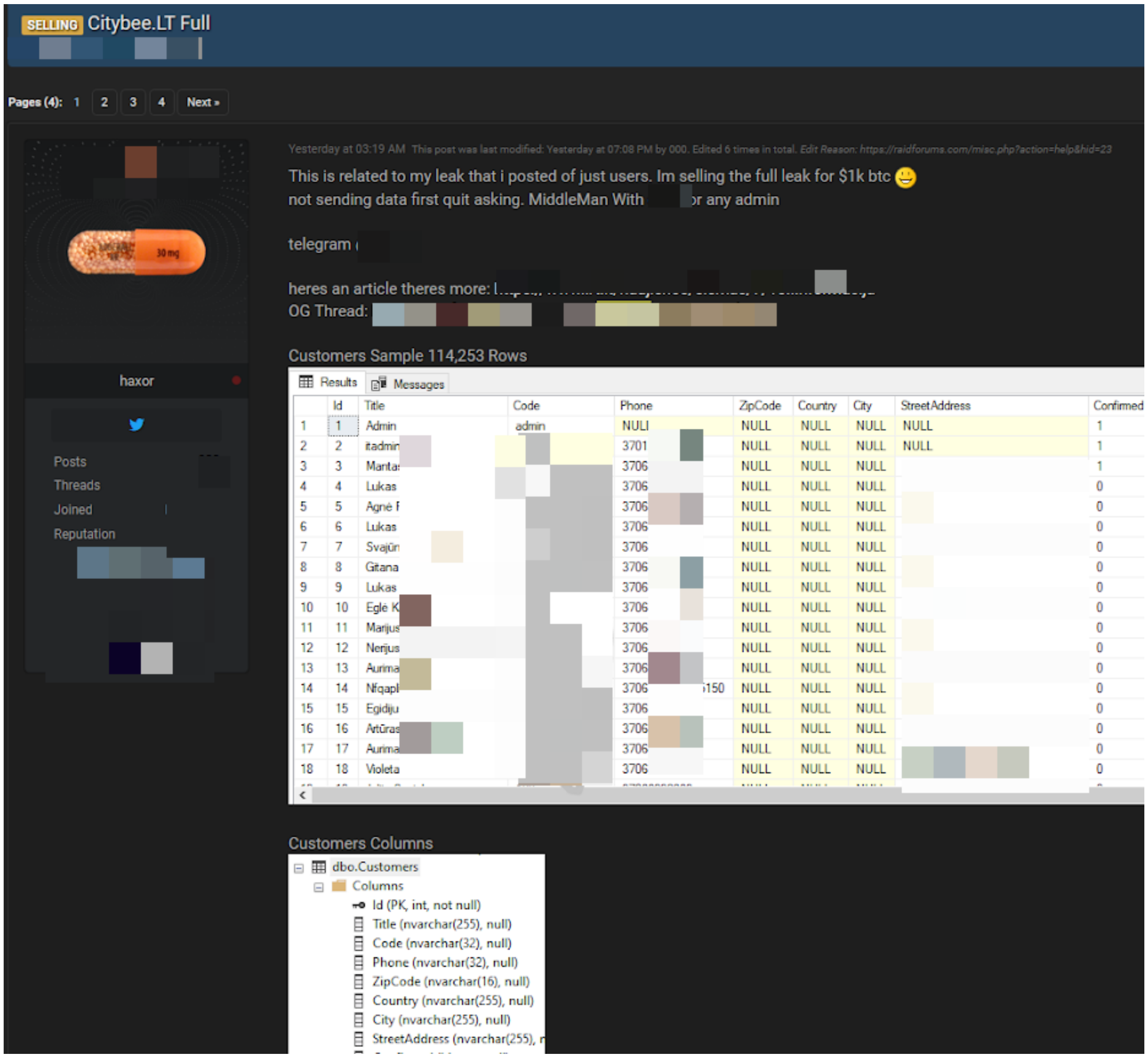
2.2 Phizer

Pfizer Inc., the world's largest pharmaceutical company, has experienced a massive data breach, with patient information discovered exposed on unprotected cloud storage.

The exposed data was discovered in a Google Cloud storage bucket that had been misconfigured. Hundreds of discussions between Pfizer's automated customer service software and consumers who used the company's prescription pharmaceuticals, including Lyrica, Chantix, Viagra, and cancer medicines Ibrance and Aromasin, were included in the data. Along with sensitive medical information, the transcripts included full names, home addresses, and email addresses, which hackers could use to conduct highly effective phishing attempts against victims: Pfizer suffers huge data breach on unsecured cloud storage. It is evident that data storage in the public cloud has become the standard, and businesses of all sizes now face complicated identity and data management challenges. Capital One demonstrated, and Pfizer has confirmed, that even with the largest teams, funds, and skill sets, the public cloud is highly difficult. When corporate organizations move at the speed of the cloud and innovate at a breakneck pace, errors and data exposure are inevitable if the necessary technologies are not in place.

2.3 Citybee

Lithuanian police were investigating after 110,000 people's personal information was exposed to an internet hacker website. CityBee, a car-sharing service, revealed that the breach compromised the records and information of thousands of its clients. euronews.com - Thousands of CityBee users have their personal data leaked online. As the attacker stated, "CityBee was using a service provided by Microsoft called Azure Blob, which is used as storage of some sorts. Now Microsoft allows you to secure those blobs with authentication, which Citybee for some reason chose not to." He was able to search CityBee in a DNS record called CNAME which linked to their azure blob and other things like their website. Even a step-by-step instruction was provided by the hacker. Essentially, it was a BACPAC file named CitybeeProduction, which contained the metadata and data from the database. Citybee also used a very weak SHA-1 encryption algorithm for passwords without any salt added.


SELLING Citybee.LT Full [redacted]

Pages (4): 1 2 3 4 Next »

haxor

Posts [redacted]
Threads [redacted]
Joined [redacted]
Reputation [redacted]

Yesterday at 03:19 AM This post was last modified: Yesterday at 07:08 PM by 000. Edited 6 times in total. *Edit Reason: https://raidforums.com/misc.php?action=help&hid=23*

This is related to my leak that i posted of just users. Im selling the full leak for $1k btc 😃
not sending data first quit asking. MiddleMan With [redacted] or any admin

telegram [redacted]

heres an article theres more: [redacted]
OG Thread: [redacted]

Customers Sample 114,253 Rows

Results | Messages

	Id	Title	Code	Phone	ZipCode	Country	City	StreetAddress	Confirmed
1	1	Admin	admin	NULL	NULL	NULL	NULL	NULL	1
2	2	itadmin [redacted]	[redacted]	3701 [redacted]	NULL	NULL	NULL	NULL	1
3	3	Manta [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	1
4	4	Lukas [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
5	5	Agnė [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
6	6	Lukas [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
7	7	Svajūn [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
8	8	Gitana [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
9	9	Lukas [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
10	10	Eglė K [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
11	11	Marijus [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
12	12	Nerijus [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
13	13	Aurima [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
14	14	Nfqapl [redacted]	[redacted]	3706 [redacted] 150	NULL	NULL	NULL	[redacted]	0
15	15	Egidiju [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
16	16	Artūras [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
17	17	Aurima [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0
18	18	Violeta [redacted]	[redacted]	3706 [redacted]	NULL	NULL	NULL	[redacted]	0

Customers Columns

dbo.Customers
Columns
Id (PK, int, not null)
Title (nvarchar(255), null)
Code (nvarchar(32), null)
Phone (nvarchar(32), null)
ZipCode (nvarchar(16), null)
Country (nvarchar(255), null)
City (nvarchar(255), null)
StreetAddress (nvarchar(255), n

On November 29, 2021, the Lithuanian data protection authority (VDAI) imposed a fine of EUR 110 000 on UAB Prime Leasing, which manages the short-term car rental platform "CityBee" (Company). VDAI found that it had failed to ensure the security of the processing of personal data.

3 Public Cloud

3.1 Cloud Security Overview

Cloud security is a critical matter. Most companies worry that highly sensitive data and intellectual property may be exposed through accidental leaks or due to increasingly sophisticated cyber attacks. Gartner predicts that through 2025, 99% of cloud security failures will be the customer's fault.

Moreover, having a solid cloud security stance helps organizations achieve other benefits, such as:

- Lower costs
- Reduced ongoing operational and administrative expenses
- Scalability
- Increased reliability and availability

- DevOps way of working

Despite bringing many benefits, the cloud computing paradigm imposes serious concerns in terms of security and privacy, which are considered hurdles in the adoption of the cloud at a very large scale (Alghofaili et al., 2021). Security issues are depended on the cloud provider, service user, instance (Sun et al., 2014), and the delivery model, PaaS, IaaS, and SaaS (Sun, 2018). Data stored in the public cloud would face both outside attacks and inside attacks (Shi, 2018). Data loss and leakage were the biggest security concern, with 44% of organizations seeing data loss as one of their top three focus areas. Two-thirds of organizations leave back doors open to attackers leading to an accidental exposure through misconfiguration. Security gaps in misconfigurations were exploited in 66% of attacks (Sophos, 2020).

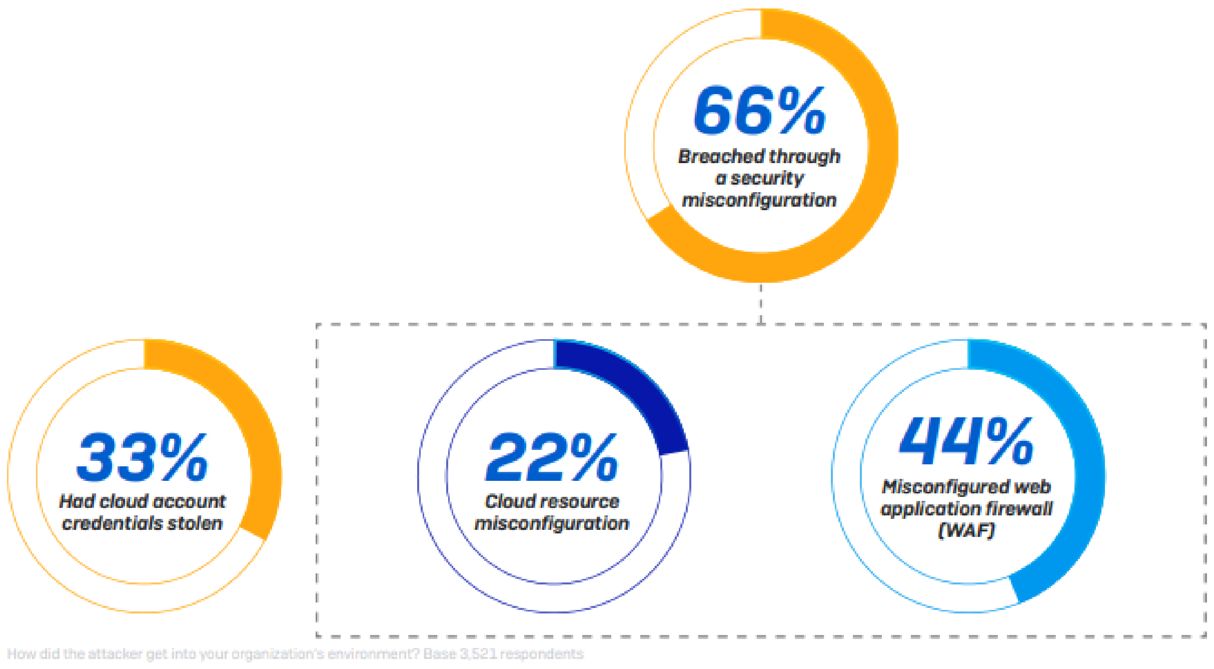


Figure 8: How criminals are getting in, source - sophos.com

Zero Trust security model enables securing cloud-native applications by encrypting all network communication, authenticating, and authorizing every request. The traditional trust management mechanisms represent a static trust relationship that falls deficit while meeting up the dynamic requirement of cloud services. (Mehraj and Banday, 2020). In order to achieve a true zero-trust security model in the cloud, a combination of network and identity permission policies should be in place.

To adequately address the modern dynamic threat environment requires(Agency, 2021):

- Coordinated and aggressive system monitoring, system management, and defensive operations capabilities.
- Assuming all requests for critical resources and all network traffic may be malicious.
- Assuming all devices and infrastructure may be compromised.
- Accepting that all access approvals to critical resources incur a risk

Some security recommendations for network security can be summarized as follows (Alghofaili et al., 2021):

- Secure communication techniques should be adopted: HTTPS for web applications, transmission channel must be encrypted by TLS
- Additional monitoring should be done (manual, automatic, ML based)
- Other public security services such as web application firewalls (WAF), virtual firewalls, virtual bastion machines, virtual host protection, and virtual database audit systems could be used

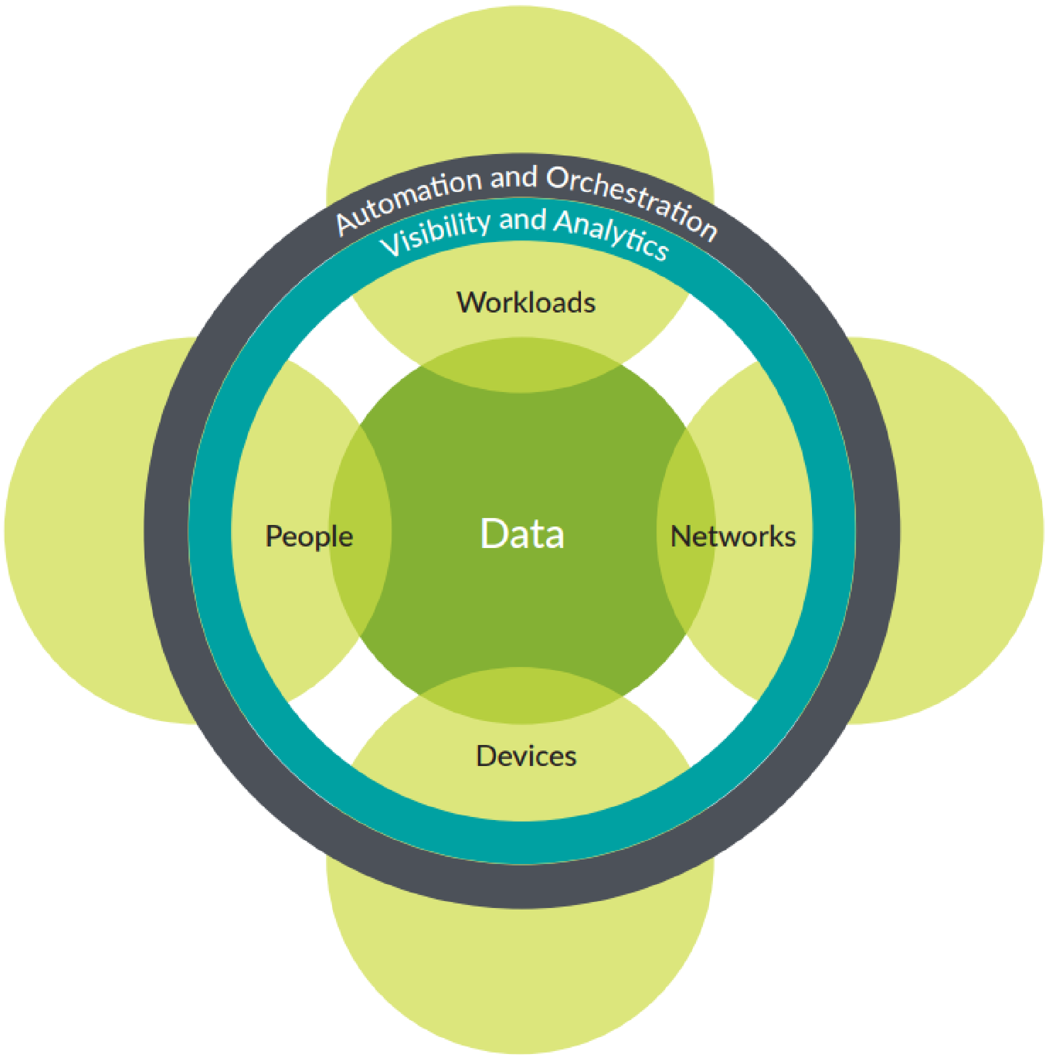


Figure 9: The Zero Trust eXtended (ZTX) Ecosystem, Forrester Research, Inc., source - juniper.net

3.2 Infrastructure as Code

There was a significant shift in development, deployment, and software application management during the past decade. The new approach is called Development Operations (DevOps) where Infrastructure as Code (IaC) plays a core role. While manual configurations in the Cloud context was a norm, nowadays it is fully automated using blueprints that are easily interpretable by machines. Moreover, IaC approach allows a faster and homogeneous configuration for the whole infrastructure. Usually, it is utilized by a specific declarative language (TerraForm, CloudFormation, Puppet) that allows users to describe the desired state of the infrastructure. This significantly reduces the time, complexity and helps to provision the infrastructure from the security, management, and costs perspectives.

The whole idea behind IaC is simple - developers can write declarative statements that define the infrastructure necessary to run the code as opposed to writing a ticket/creating a task for administrators. Reproducibility and transparency come as a side effects.

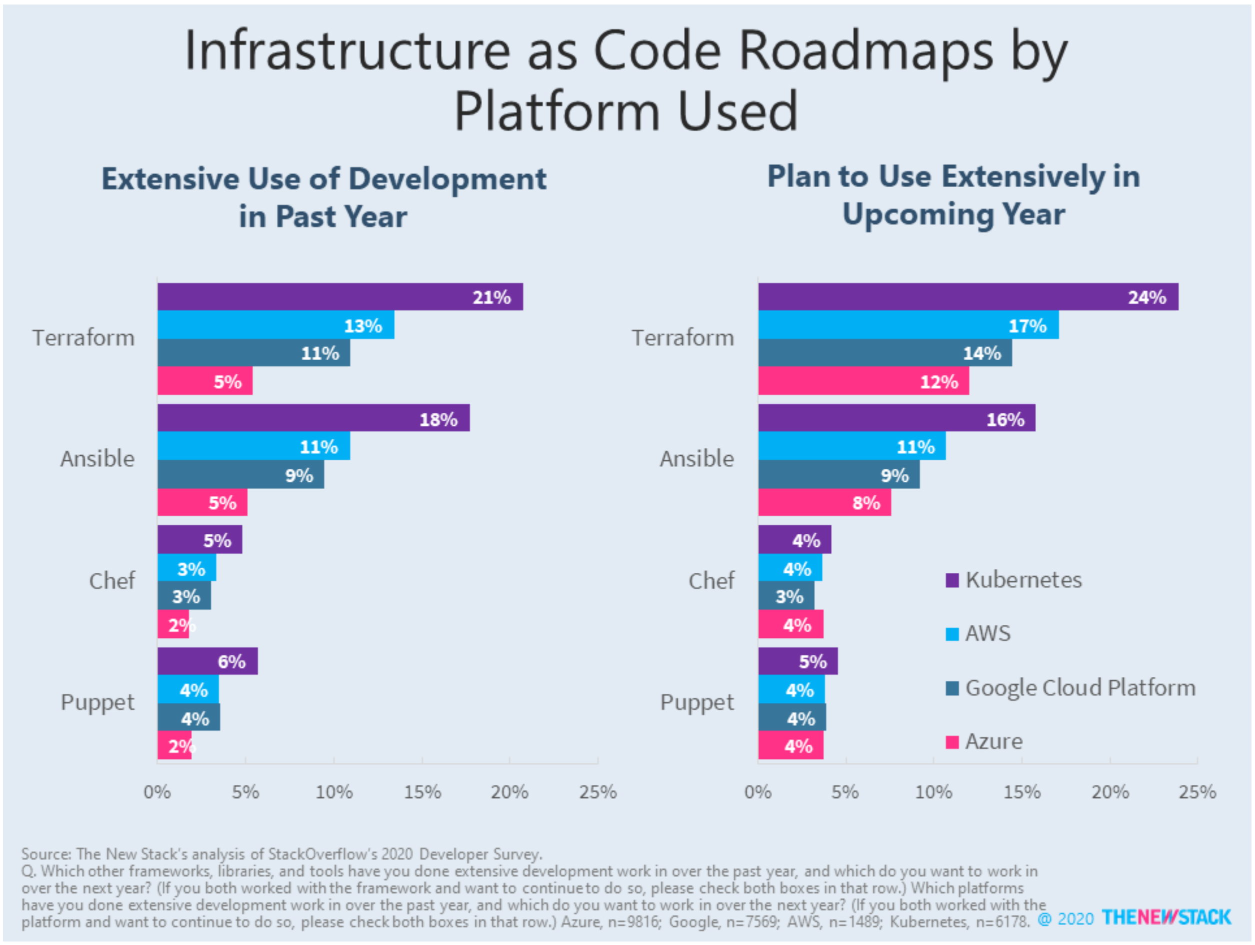


Figure 10: Infrastructure as Code Survey, source - thenewstack.io

Terraform is one of the most popular ways to implement this pipeline, especially in a Cloud context. It is an open-source tool that lets you provision Google Cloud resources with declarative configuration files-resources such as virtual machines, containers, storage, and networking. It lets users manage Terraform configuration files in source control to maintain an ideal provisioning state for testing, production, and other environments. (Almuairfi and Alenezi, 2020)

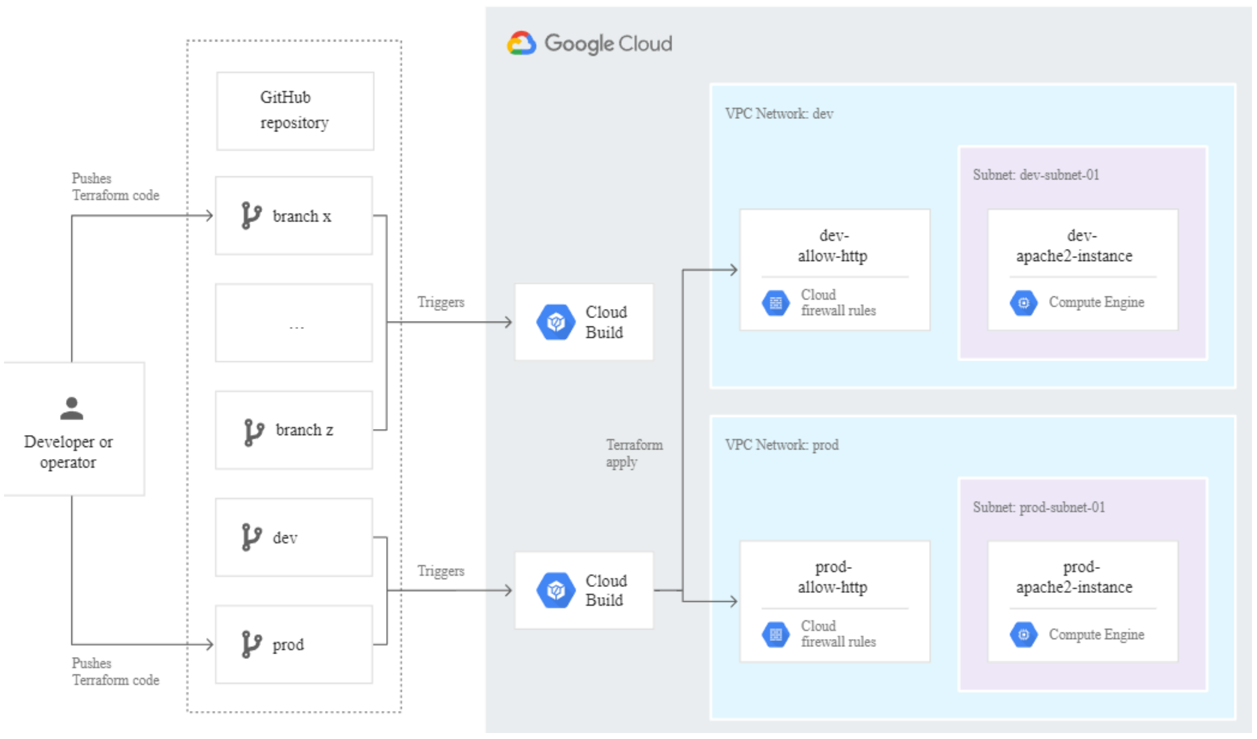


Figure 11: Terraform example, source - cloud.google.com

4 Project Scope

4.1 Platform for anatical applications

Shiny Server is a powerful open source back end application. It creates a web server optimized for hosting Shiny applications. Shiny Server enables you to host your apps in a controlled environment, such as within your corporation, ensuring that your Shiny app (and any data it requires) never leaves your control.

The majority of Shiny apps are deployed within an organization's firewall, and because you can generally trust that your colleagues will not attempt to hack your app64, you don't need to worry about security. If, on the other hand, your app contains data that only a subset of your colleagues should have access to, or if you wish to make your app publicly accessible, you will need to invest some effort in security. When it comes to app security, there are two critical components to safeguard: your data and your compute resources.

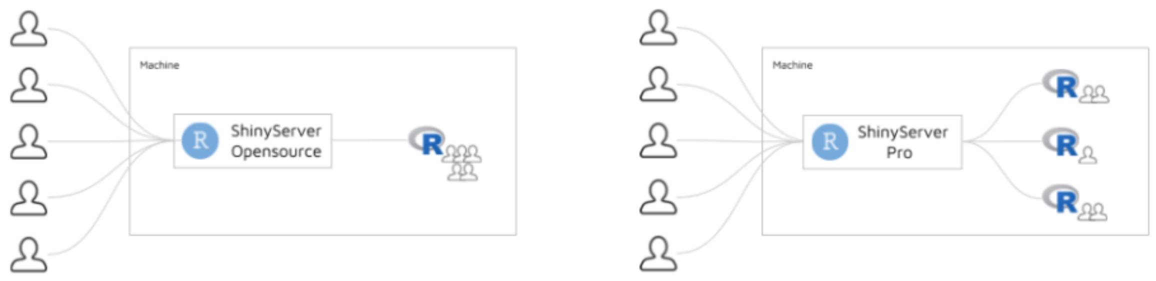


Figure 12: Shiny Server vs Shiny Server Pro

While Shiny Server has some security features, the open source version is pretty basic: it does not have authentication, TLS/SSL (traffic is not encrypted), there's no defense to a denial of service attack and there is no way to distribute work for multiple R instances. However, some companies, e.g. Appsilon are specializing in enterprise Shiny dashboards using the open source version. We will attempt to establish a similar secure infrastructure in this project and to expose shiny applications to the world wide web as well.

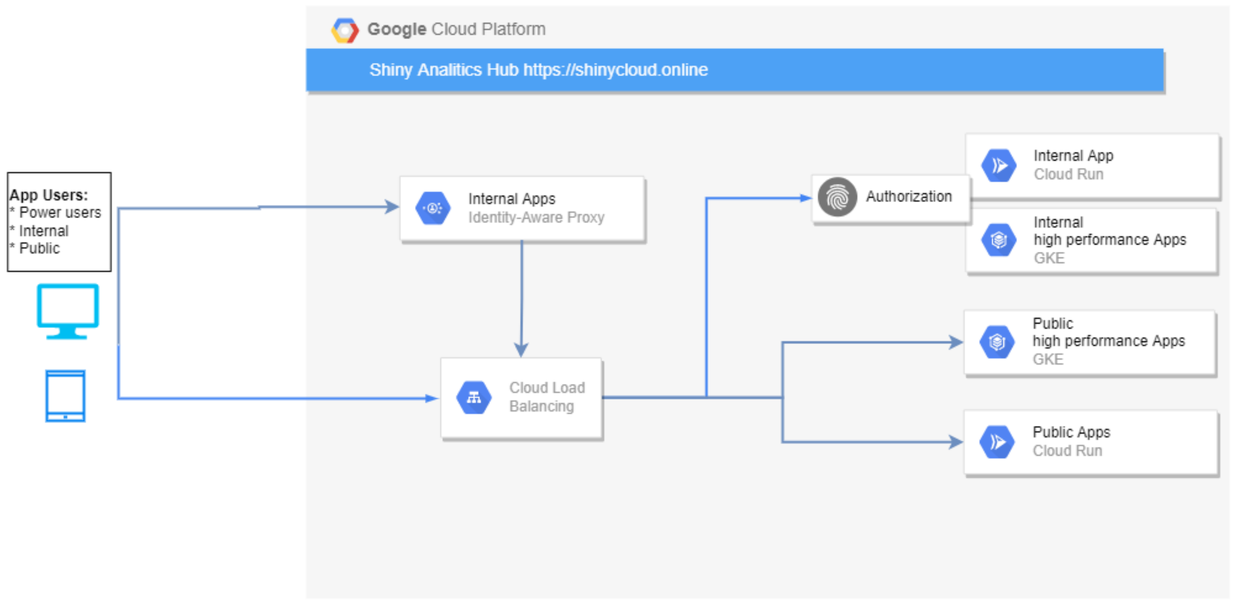


Figure 13: Shiny Analytics Hub - Architecture

4.2 Kubernetes and Docker

The Open Source Shiny Server is a stand-alone application that runs on a single thread. While it is feasible to run many instances of shiny servers and divide work via load balancing, this is inefficient and far more expensive than using containerized applications. While it is possible to create the shiny-server container from scratch, there are a lot of pre-build docker images available including all the dependencies. Moreover, virtualized applications are highly preferred in general as opposed to IaaS approach (virtual machines), it makes sense to dig deeper in kubernetes and docker setup, regardless of the chosen service provider.

Kubernetes was founded by Ville Aikas, Joe Beda, Brendan Burns, and Craig McLuckie in collaboration with Google engineers Brian Grant and Tim Hockin in mid-2014. Google's Borg system heavily influenced kubernetes design (Verma et al., 2015) (Burns et al., 2016). While the Borg project was implemented entirely in C++, Kubernetes was rewritten in Go language. The main goal of kubernetes was to build on the capabilities of containers and provide significant gains in programmer productivity while easing the management of the system.

Kubernetes is the most popular container orchestration platform that enables users to create and run multiple containers in cloud environments. Kubernetes offers resource management to isolate the resource usage of containers on a host server because performance isolation is an important factor in terms of service quality.

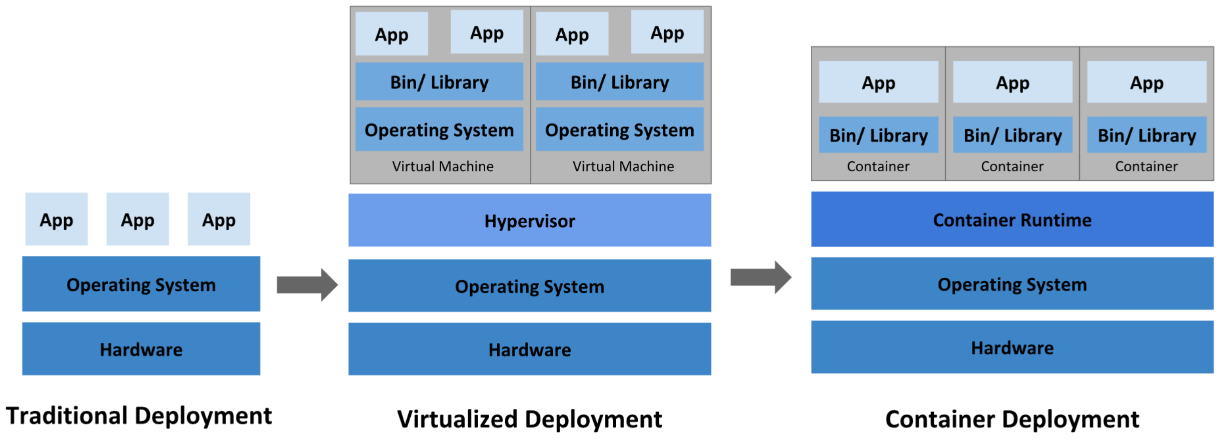


Figure 14: Container evolution - kubernetes.io

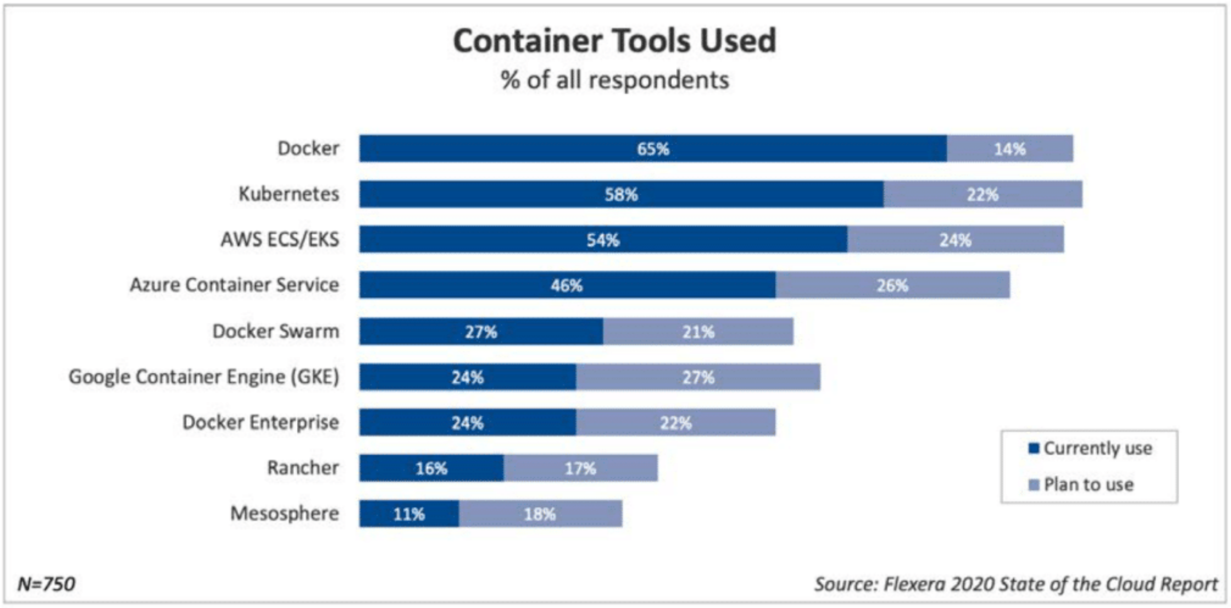
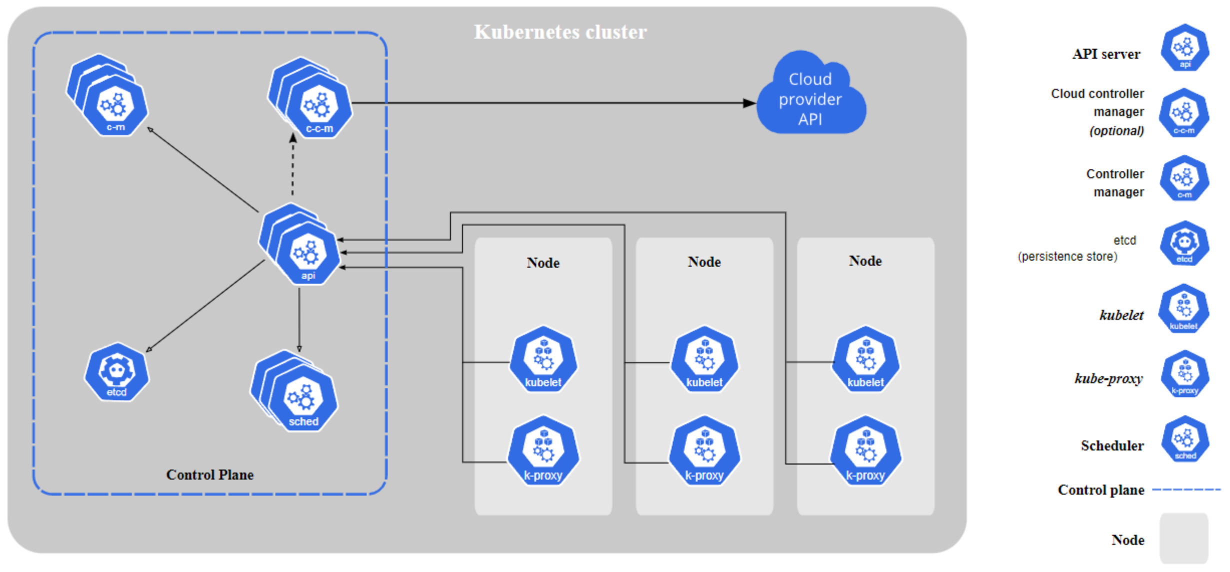


Figure 15: Cloud Strategy - image source Flexera



Environments running Kubernetes consist of the following key components:

- Kubernetes control plane—manages Kubernetes clusters and the workloads running on them. Include components like the API Server, Scheduler, and Controller Manager.
- Kubernetes data plane—machines that can run containerized workloads. Each node is managed by the kubelet, an agent that receives commands from the control plane.
- Pods—pods are the smallest unit provided by Kubernetes to manage containerized workloads. A pod typically includes several containers, which together form a functional unit or microservice.
- Persistent storage—local storage on Kubernetes nodes is ephemeral, and is deleted when a pod shuts down. This can make it difficult to run stateful applications. Kubernetes provides the Persistent Volumes (PV) mechanism, allowing containerized applications to store data beyond the lifetime of a pod or node.

4.3 Shiny Server without security

Computer networks are prone to attacks and it has a wide range of attacks associated with them. Cloud is not an exception and even holds more risk. It can be prone to Denial-of-service, Eavesdropping, Host Attacks, Password Guessing, Protocol-based, and Social Engineering attacks (Chopra, 2016).

As an experiment, the firewall was opened to the whole world and network activity was monitored for one week. While the activity in the Compute Instance (Shiny Server hosted on a Virtual Server) was marginal, the exposed Shiny Server instance on Google Kubernetes Cluster was scanned extensively. This could be due to the rules on how Google generates IP addresses for corresponding instances. Moreover, GKE was exposed on port 80 which is a standard HTTP port, while the standard port of shiny server (3838) was used for Compute Instance, which is not that common configuration.

While the majority of the requests came from the USA, applications from China and Russia also scanned our exposed application considerably. These scans also are not centralized but are rather done by individuals or companies which specialize in data mining and web crawling. Some requests are also received from Lithuania, CGates Internet Service Provider.

Some of the IPs were crossed check with a publicly available IP database. These IP addresses, especially from China and Russia, were already reported a number of times and are indicated as abusive.

The analysis proves that an incorrectly configured firewall poses one of the most significant security risks. Misconfigured applications could serve as a back door and is a low handing fruit for hackers - e.g. it is easy to run a port scan for a specific IP range and use a collection of scripts/exploits to check whether there are any holes in the application. If any sensitive data where General Data Protection Regulation is not applied

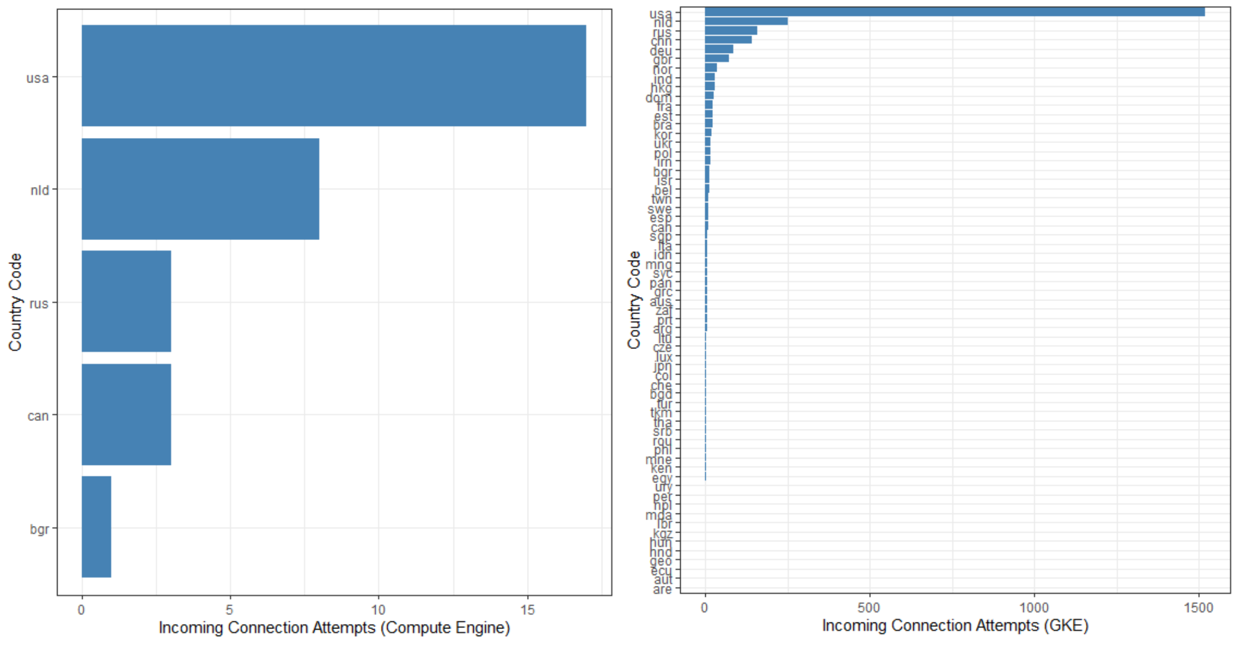


Figure 16: Incoming Requests

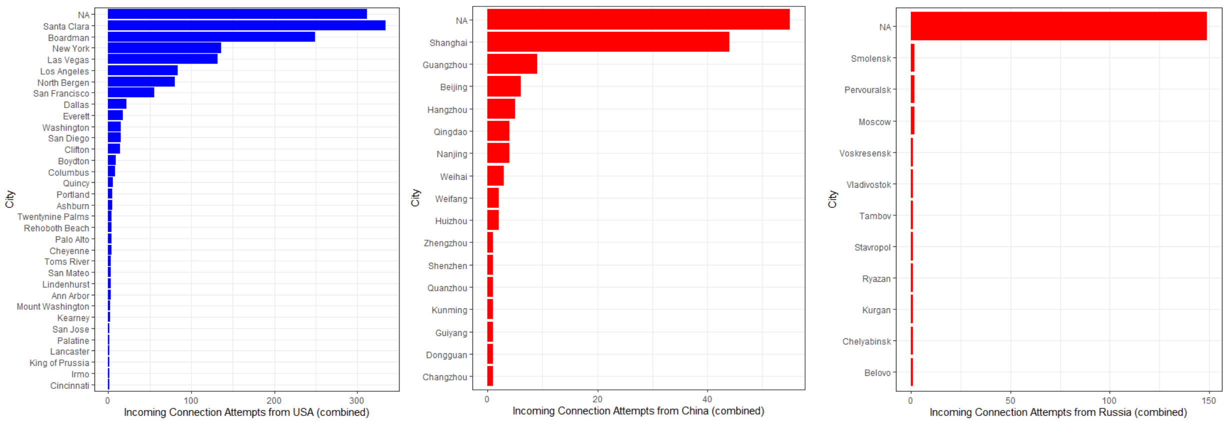
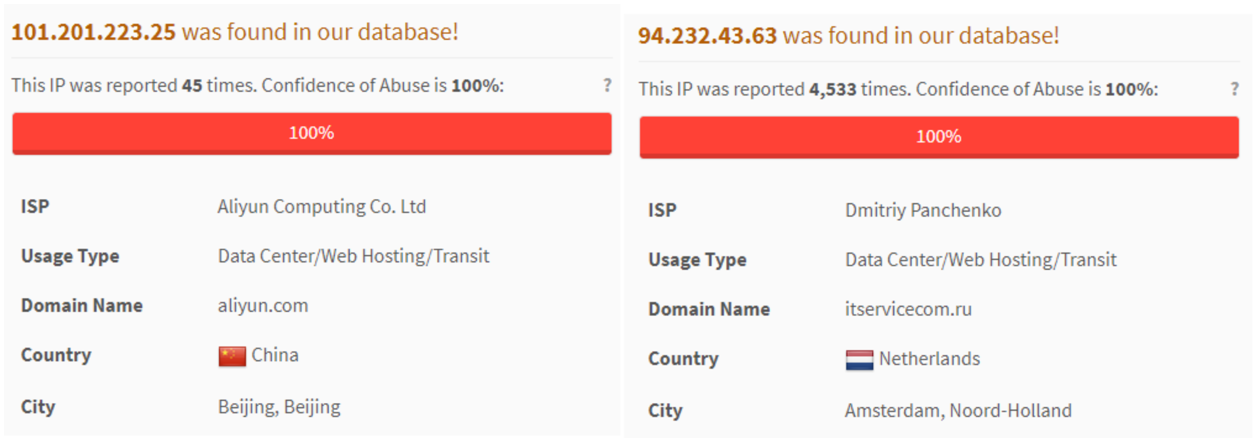


Figure 17: Incoming Requets from different cities



101.201.223.25 was found in our database!

This IP was reported **45** times. Confidence of Abuse is **100%**: ?

100%

ISP	Aliyun Computing Co. Ltd
Usage Type	Data Center/Web Hosting/Transit
Domain Name	aliyun.com
Country	China
City	Beijing, Beijing

94.232.43.63 was found in our database!

This IP was reported **4,533** times. Confidence of Abuse is **100%**: ?

100%

ISP	Dmitriy Panchenko
Usage Type	Data Center/Web Hosting/Transit
Domain Name	itservicecom.ru
Country	Netherlands
City	Amsterdam, Noord-Holland

Figure 18: Blacklisted IPs - source https://www.abuseipdb.com/

(i.e. USA, China, Russia).

5 Security considerations

5.1 Data Encryption

TLS (Transport Layer Security) encrypts data transmitted over the Internet, ensuring that snoopers and hackers cannot view what you transmit. This is especially important for private and sensitive information such as passwords, credit card numbers, and personal communications. TLS originated from Secure Socket Layers (SSL), which was originally created to secure web sessions in 1994 by Netscape Communications Corporation. SSL 1.0 was never released publicly, while SSL 2.0 was rapidly superseded by SSL 3.0, the foundation for TLS. Historically, data was transferred unencrypted over the Internet, and encryption was often applied piecemeal to critical information such as passwords or payment information. Without TLS, not only may sensitive information such as usernames, passwords, and personal information be easily gathered, but also surfing habits, e-mail conversations, online chats, and conference calls can be observed. By enabling TLS support in client and server apps, it ensures that data exchanged between them is encrypted using secure algorithms and is inaccessible to third parties.

TLS is the most frequently used encryption protocol in the internet. It is composed of two protocols: the TLS Handshake Protocol, which is responsible for authentication and key establishment, and the TLS Record Protocol, which is responsible for the subsequent usage of those keys to secure bulk data. SSL/TLS protects the confidentiality and integrity of data-in-transit by utilizing both asymmetric and symmetric encryption. Asymmetric encryption is used to establish a secure connection between a client and a server, and symmetric encryption is used to exchange data within the secured connection.

At a high level, TLS is split into two phases:

- A handshake phase where a secure communication is negotiated and created between two participants.
- A post-handshake phase where communications are encrypted between the two participants.

TLS 1.2 and earlier versions perform the key exchange once both parties agree on the key exchange algorithm to use. This means they first agree on a method to utilize before exchanging public keys.

To eliminate the first round of negotiation (one client message and one server message) in TLS 1.3, the client transmits a public key speculatively in the very first message (the Client Hello). If the client is unable to correctly forecast the server's key exchange mechanism, the client must send a new Client Hello with the right public key. TLS 1.3 contains numerous such enhancements that are critical for the web. Indeed, many people have insecure or poor connections, making it critical to restrict non-application communication to a bare minimum. Additionally, unlike prior versions of TLS, all key exchanges in TLS 1.3 are ephemeral. This

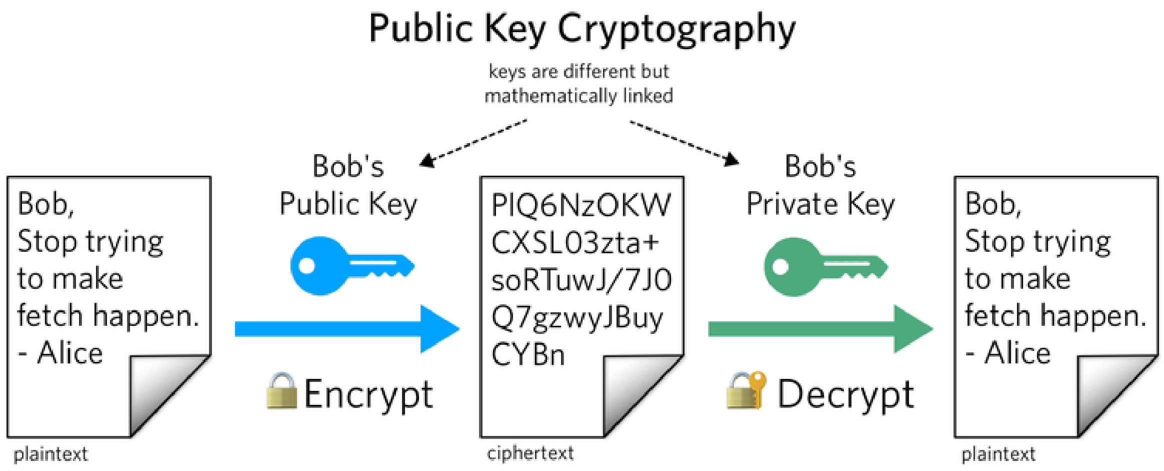

Public Key Cryptography
keys are different but mathematically linked
Bob's Public Key
Bob's Private Key
Bob, Stop trying to make fetch happen. - Alice
plaintext
Encrypt
PIQ6NzOKW CXSL03zta+ soRTuwJ/7J0 Q7gzwyJBuy CYBn
ciphertext
Decrypt
Bob, Stop trying to make fetch happen. - Alice
plaintext

Figure 19: Asymmetric (Public Key) Cryptography use case as encryption - source https://www.twilio.com/blog/what-is-public-key-cryptography

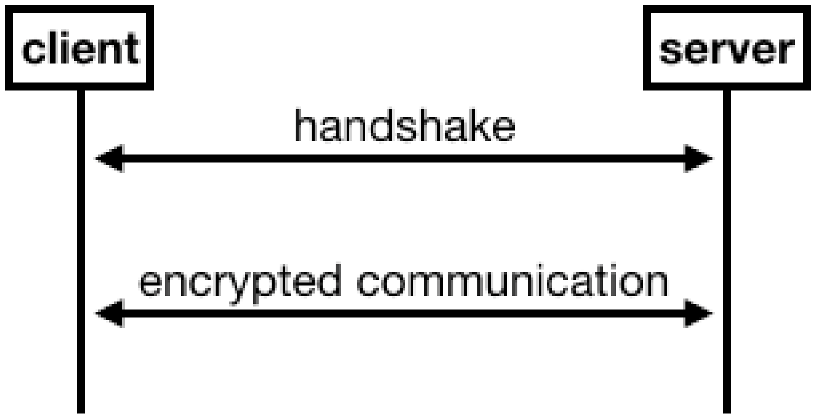


Figure 20: A handshake and post-handshake communitcation

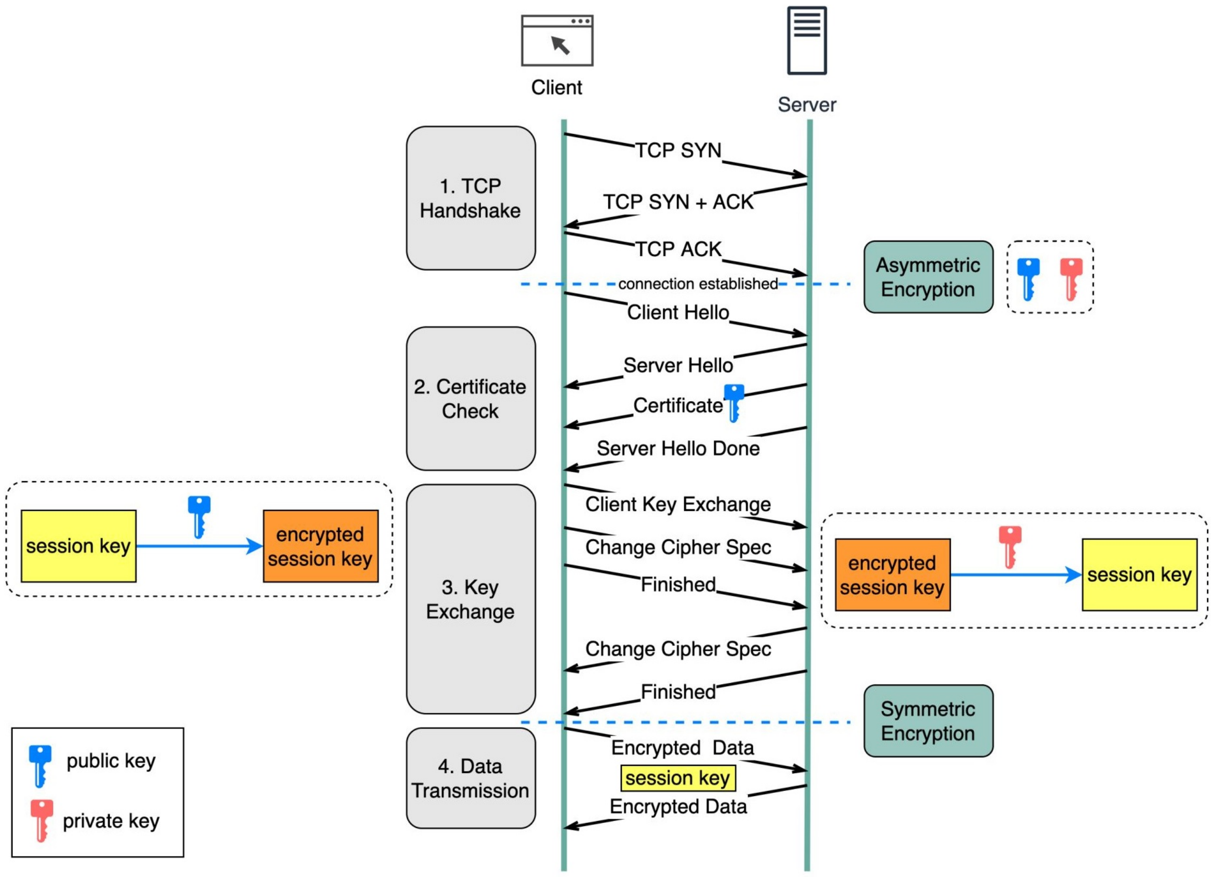


Figure 21: Asymmetric and symmetric within TLS - source https://blog.bytebytego.com/

means that for each new session, both the client and server generate new key pairs and then delete them immediately after the key exchange is complete (Wong, 2021).

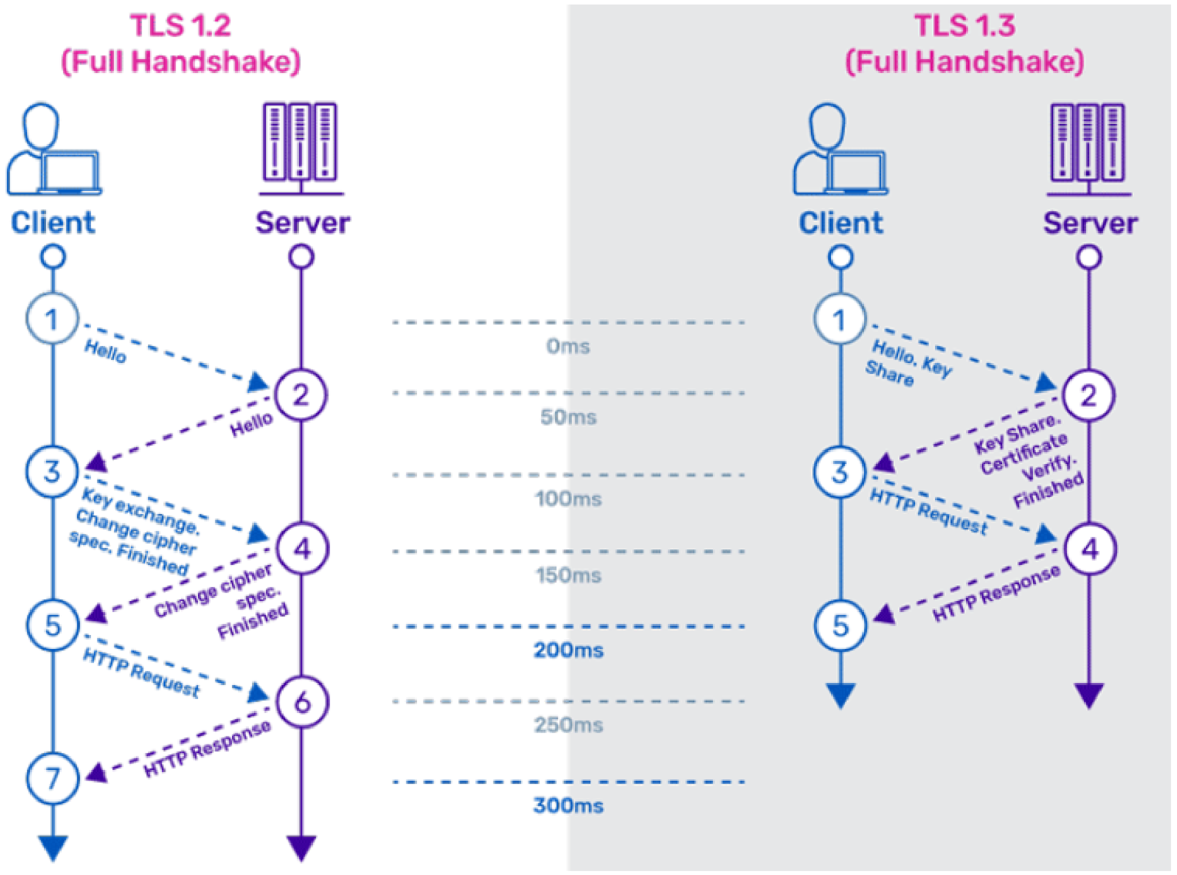


Figure 22: TLS performance comparison - https://www.a10networks.com

The handshake itself contains of these following steps:

- Negotiation. TLS is highly configurable. Both a client and a server can be configured to negotiate a range of SSL and TLS versions, as well as a menu of acceptable cryptographic algorithms. The negotiation phase of the handshake aims at finding common ground between the client's and the server's configurations, in order to securely connect the two peers.
- Key exchange. The whole point of the handshake is to perform a key exchange between the two participants. What key exchange algorithm to use? This is one of the things decided as part of the negotiation process.
- Authentication. It is trivial for a MITM attacker to impersonate any side of a key exchange. For this reason, key exchanges must be authenticated. (Your browser must have a way to make sure that it is talking to google.com and not your Internet service provider, for example.)
- Session Resumption. As browsers often connect to the same websites again and again, key exchanges can be costly and slow down a user's experience. For this reason, mechanisms to fast-track secure sessions without redoing a key exchange are integrated into TLS.

An example of TLS 1.3 a negotiation could be:

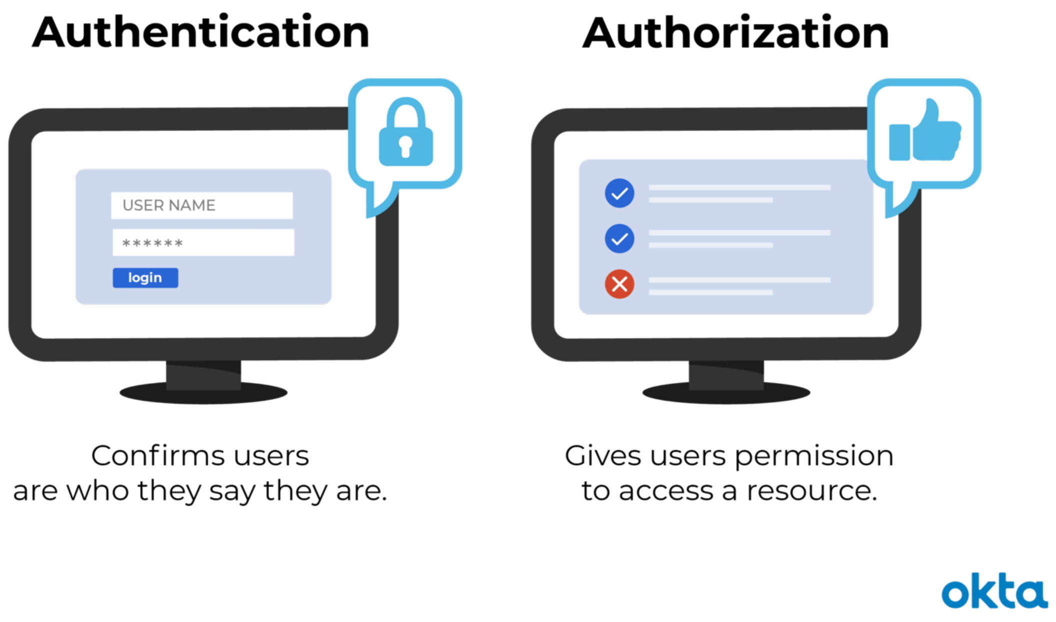
```
* The client sends a TLS 1.3 Client Hello announcing that it can do either an
* X25519 or an X448 key exchange together with a X25519 public key.
* The server does not support X25519, but does support X448.
* It sends a Hello Retry Request to the client announcing that it only supports X448.
* The client sends the same Client Hello but with an X448 public key instead.
* The handshake goes on.
```

RSA and the Diffie-Hellman Key Exchange are the two most widely used encryption algorithms, each of which solves the same problem in a somewhat different way. In a nutshell, the Diffie Hellman technique generates a public and private key for each party to the transaction but shares only the public key. Unlike Diffie-Hellman, the RSA algorithm can be used for both digital signatures and symmetric key exchange, although it does need the prior exchange of a public key. RSA stands for Ron Rivest, Adi Shamir, and Leonard Adleman—the three men who initially publicly defined the approach in 1977. RSA is based on the assumption that factoring really large integers is difficult (integer factorization). On the assumption that there is no efficient solution for integer factorization, it is thought that decrypting an RSA ciphertext completely is infeasible. A user of RSA generates and then publishes their public key, which is composed of two enormous prime integers and an auxiliary value. The primary variables must remain unidentified. While anyone can use the public key to encrypt a message, only someone familiar with the prime factors is likely to be able to decrypt it.

The browser, on the other hand, verify that the handshake is happening between the client and specific website using the Public Key Infrastructure. Firstly, browsers must trust a set of root public keys - Certificate Authorities (CAs). Typically, browsers will either utilize a hardcoded list of trusted public keys or will request them from the operating system. Websites wishing to utilize HTTPS must be able to receive certification from one of these CAs (a signature of their signing public key). In order to do this, a website owner must prove to a CA that they own a specific domain. CAs do not directly sign public keys but rather sign certificates. A certificate contains both the long-term public key and some crucial metadata, such as the domain name of your web page, if you have one. To ensure that the server connecting with a browser is genuine, the server includes a certificate chain in the TLS handshake. This certificate chain is sent in a Certificate TLS message by the server, and by the client as well if the client has been asked to authenticate. The server sends this certificate chain in a Certificate TLS message, as does the client if the client is prompted to authenticate. Following the establishment of a handshake and the generation of symmetric keys, both the client and server can exchange encrypted application data. Additionally, TLS prevents reordering of such communications.

5.2 Single Sign On

Authentication and trust mechanisms are needed by the user and provider alike. In this scenario, SSO could be a good starting point. The spam e-mail problem can be also mitigated in the cloud. Despite their frequent interchangeability, authentication and permission serve fundamentally distinct purposes. Authentication is the process of validating a user's identity, whereas authorization is the process of checking the user's access rights. As a general recommendation, the SSO (Single Sign-on) and CAS (Central Authentication Service) should be also activated to make a user to access various software services through a single active user account. (Jangjou and Sohrabi, 2022)


Authentication
Authorization
USER NAME
******
login
Confirms users are who they say they are.
Gives users permission to access a resource.
okta

SAML and OAUTH2 are two of the most popular Single Sign-On implementation methods.

5.2.1 SAML

SAML is an abbreviation for Security Assertion Markup Language (SAML). Its principal function in terms of online security is to permit access to different web applications with a single set of login credentials. It operates by exchanging authentication data between two parties, often an identity provider (idP) and a web application. Web applications utilize SAML to send authentication data between two parties - the identity provider (IdP) and the service provider - in an XML-based format (SP). SAML was developed by the technology sector to ease the authentication procedure for users requiring access to various, independent web apps across domains. Single sign-on (SSO) was possible before SAML, but it relied on cookies that were only valid within the same domain. It accomplishes this by centralizing user authentication through an identity provider. Then, Web apps can utilize SAML via the identity provider to provide their users access. This SAML authentication method eliminates the need for users to memorize numerous login credentials. It also benefits service providers because it enhances the security of their own platform, especially by eliminating the need to retain (often insecure and weak) passwords and eliminating forgotten password difficulties. SAML V1.0 became an OASIS standard in November 2002. SAML V1.1 followed in September 2003 and has seen significant success, gaining momentum in financial services, higher education, government, and other industry segments. SAML is an important standardized example of this new protocol class and will be widely used in business-to-business scenarios to reduce user-management costs. (Nongbri et al., 2018).

Even though SSO reduces the number of logins that are needed over heterogeneous environments, the risk that might be associated with the security of SSO might be detrimental if, for example, a Man-in-the Middle (MITM) attacker manages to gain control of the SSO credentials. The problem that a paper addresses is the lack of a proactive technique of hardening cloud-based SAML while combining SSO with a Multi-Factor Authentication (MFA) (Karie et al., 2020).

with Redirect/POST Bindings and to the protocol which is distributed and used by Google's partner companies to get full control over the authorization and authentication of hosted user accounts that can access web-based applications like Gmail or Google Calendar (Armando et al., 2008)

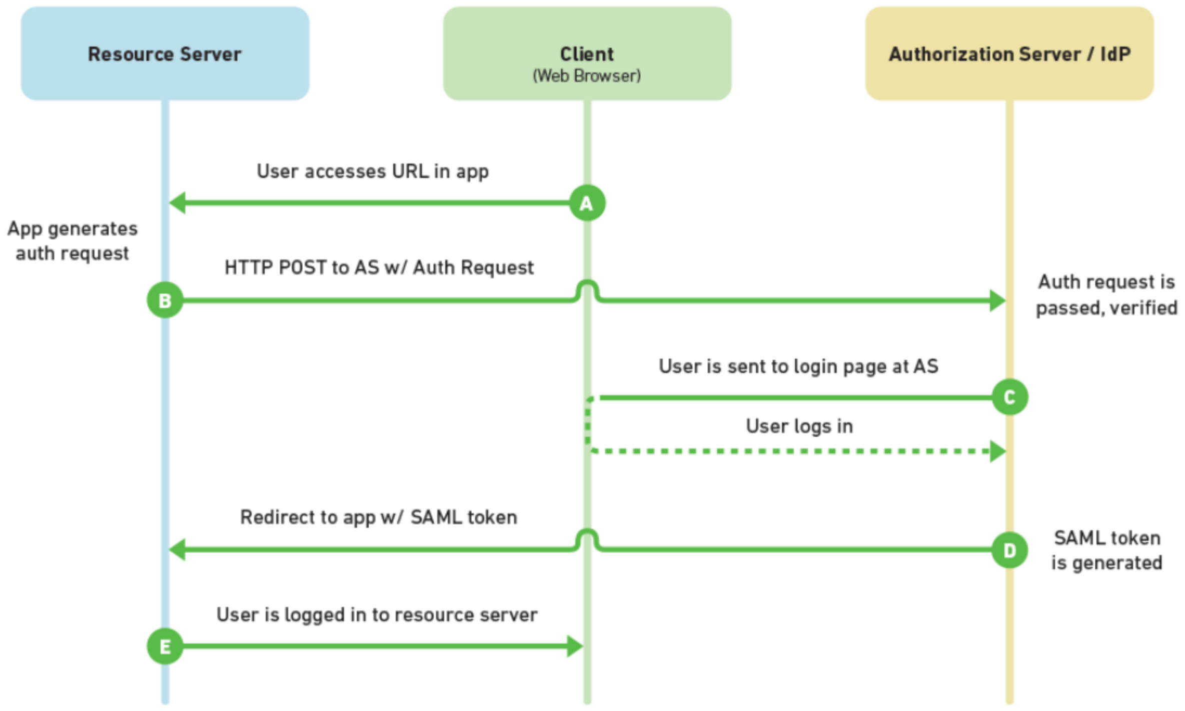

Resource Server
Client
(Web Browser)
Authorization Server / IdP
User accesses URL in app
A
App generates auth request
HTTP POST to AS w/ Auth Request
B
Auth request is passed, verified
User is sent to login page at AS
C
User logs in
Redirect to app w/ SAML token
D
SAML token is generated
User is logged in to resource server
E

Figure 23: SAML 2.0 Flow - source www.mutuallyhuman.com

SAML, however, provides a feature that OAuth2 does not: the SAML token contains user identification information (because of signing). With OAuth2, this is not provided by default; instead, the Resource Server must make an additional round trip to the Authorization Server to validate the token.

5.2.2 OAuth2

OAuth 2.0 (henceforth OAuth2) is a much more recent specification than SAML (circa late 2012). It has the advantage of being recent and takes into account the changes in the world during the past eight years. OAuth2 offers a simpler and more standardized solution that satisfies all of our existing requirements and eliminates the need for interoperability workarounds with native applications.

OAuth 2.0 is one of the most extensively used authorization/single sign-on (SSO) protocols and serves as the foundation for the new SSO standard OpenID Connect. OAuth 2.0 is also popular for cloud-based online services and is used by many large companies, like Google, Facebook, and Microsoft, to allow restricted access to data. OAuth 2.0 has a flexible architecture and may be implemented and customized in a variety of ways based on the use case.

Nevertheless, each of these implementation decisions and configuration options has its own security implications. Despite OAuth's popularity, analysis attempts to date have focused primarily on discovering faults in individual implementations and have relied on formal models that abstract from numerous web aspects or have not provided a formal approach.

four major attack types are still feasible if the security recommendations and certain best practices are followed, e.g. 307 redirect attack, IdP mix-up attack, state leak attack, and naive RP session integrity attack (Fett et al., 2016). In addition to the current approach of basic authentication via client secret, it is recommended to add 2FA (Two Factor Authentication) as a feature for third-party client app authentication. (Singh and Chaudhary, 2022)

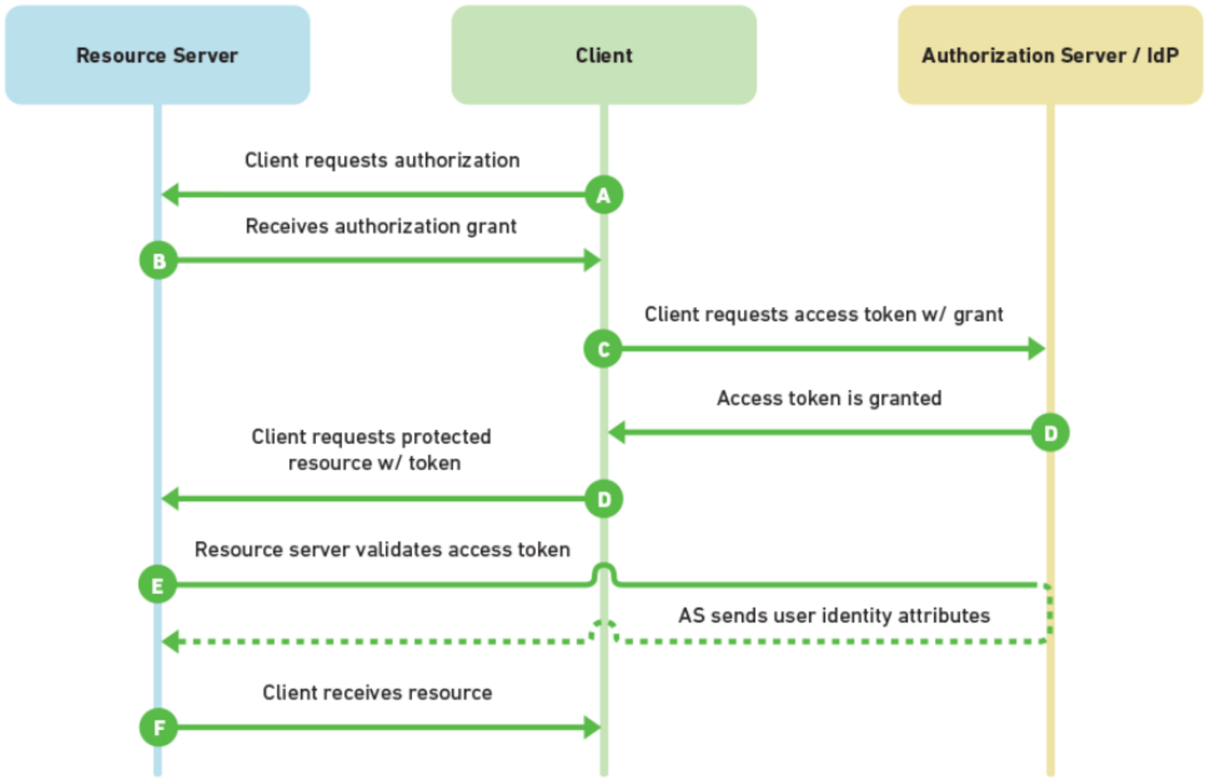

Resource Server
Client
Authorization Server / IdP
Client requests authorization
A
Receives authorization grant
B
Client requests access token w/ grant
C
Access token is granted
D
Client requests protected resource w/ token
D
Resource server validates access token
E
AS sends user identity attributes
Client receives resource
F

Figure 24: OAuth 2.0 Flow - source www.mutuallyhuman.com

5.3 Network

Providing secure services requires the setup of cloud infrastructure security policies appropriately. After gaining access to this equipment, they are able to attack the host layer's weaknesses and conduct malicious operations in this layer. Other misconfiguration vulnerabilities can be characterized as follows (Jangjou and Sohrabi, 2022):

- Activating unused ports of equipment
- Opening ports and unnecessary services such as HTTP on equipment
- No upgrade of firmware of equipment
- No periodic use of security patches
- No configuration of passwords in various work modes of equipment
- Using unsecure remote protocols such as Telnet
- No use of tunnel protocols, such as IP security, to encrypt users' accounts and intranet traffic

All configurations should be recorded, maintained, and monitored under scheduled programs to prevent any error or misconfiguration which is fairly easy to achieve using the cloud services and infrastructure as code approach.

Logging the events such as successful/unsuccessful access of the users to the equipment, configurations, and malicious traffics, and storing the logfiles on a separate server with limited access over a long period of time. Logging events such as successful/failed user access to equipment, configurations, and malicious traffic, and storing logfiles on a separate server with restricted access for an extended period of time (Jangjou and Sohrabi, 2022).

When it comes to networking, logical isolation makes a VPC environment inherently more secure. However, public cloud security isn't automatic, even with VPC – it must be intentionally deployed. Cloud security is always a shared responsibility between a cloud provider and its clients.

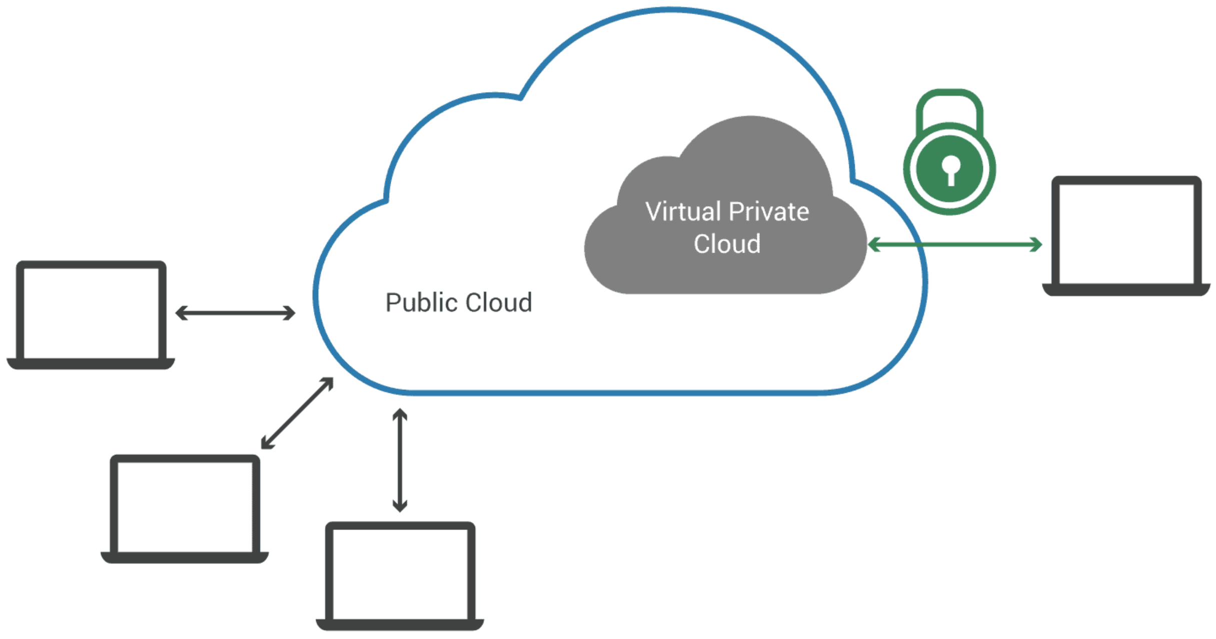


Figure 25: Virtual Private Cloud

6 [TBA] Implementation in GCP

TBA

https://shinycloud.online/

https://login.shinycloud.online/

6.1 Conteinarization

```
FROM rocker/shiny

# install R package dependencies
RUN apt-get update && apt-get install -y \
libcurl4-openssl-dev libssl-dev \
## clean up
&& apt-get clean \
&& rm -rf /var/lib/apt/lists/ \
&& rm -rf /tmp/downloaded_packages/ /tmp/*.rds

## Install packages from CRAN

RUN install2.r --error \
    -r 'http://cran.rstudio.com' \
    googleAuthR

COPY shiny-customized.config /etc/shiny-server/shiny-server.conf
COPY client.json /srv/shiny-server/client.json
COPY app.R /srv/shiny-server/app.R
RUN chmod -R 775 /srv/shiny-server/

EXPOSE 8080

USER shiny

CMD ["/usr/bin/shiny-server"]
```

```
gcloud auth login --project {project id}
PROJECTID=$(gcloud config get-value project)
docker build . -t gcr.io/$PROJECTID/signin
docker push gcr.io/$PROJECTID/signin
```

6.1.1 Cloud Run

6.1.2 GKE

6.2 Shiny App security features

6.2.1 Shiny server authentication

6.2.2 Shiny authorization

6.3 Cloud Security

6.3.1 Firewalls

6.3.2 DNS

```
gcloud beta dns --project=$PROJECTID managed-zones create shinycloud-online --description="" --dns-name
```

6.3.3 Certificates

6.3.4 Identity aware proxy

6.3.5 Audit

References

References

Agazzi, A. E. (2020). Business email compromise (BEC) and cyberpsychology. *CoRR*, abs/2007.02415.

Agency, N. S. (2021). Embracing a zero trust security model.

Alghofaili, Y., Albattah, A., Alrajeh, N., Rassam, M. A., and Al-rimy, B. A. S. (2021). Secure cloud infrastructure: A survey on issues, current solutions, and open challenges. *Applied Sciences*, 11(19).

Alkhalil, Z., Hewage, C., Nawaf, L., and Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science*, 3.

Almuairfi, S. and Alenezi, M. (2020). Security controls in infrastructure as code. *Computer Fraud and Security*, 2020:13–19.

Armando, A., Carbone, R., Compagna, L., Cuellar, J., and Tobarra, L. (2008). Formal analysis of saml 2.0 web browser single sign-on. pages 1–10.

Bairagi, S. and Bang, A. (2015). *Cloud Computing: History, Architecture, Security Issues.*

Burns, B., Grant, B., Oppenheimer, D., Brewer, E., and Wilkes, J. (2016). Borg, omega, and kubernetes. *ACM Queue*, 14:70–93.

Chopra, A. (2016). Security issues of firewall. *International Journal of P2P Network Trends and Technology*, 22:4–9.

Fett, D., Küsters, R., and Schmitz, G. (2016). A comprehensive formal security analysis of oauth 2.0. *CoRR*, abs/1601.01229.

Garg, S. and Garg, S. (2019). Automated cloud infrastructure, continuous integration and continuous delivery using docker with robust container security.

Hammouchi, H., Cherqi, O., Mezzour, G., Ghogho, M., and Koutbi, M. E. (2019). Digging deeper into data breaches: An exploratory data analysis of hacking breaches over time. *Procedia Computer Science*, 151:1004–1009. The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops.

Hlaing, Z. C. S. S. and Khaing, M. (2020). A detection and prevention technique on sql injection attacks. In *2020 IEEE Conference on Computer Applications(ICCA)*, pages 1–6.

Humayun, M., Niazi, M., Zaman, N., Alshayeb, M., and Mahmood, S. (2020). Cyber security threats and vulnerabilities: A systematic mapping study. *Arabian Journal for Science and Engineering*, 45.

Jangjou, M. and Sohrabi, M. (2022). A comprehensive survey on security challenges in different network layers in cloud computing. *Archives of Computational Methods in Engineering.*

Javeed, D. and MohammedBadamasi, U. (2020). Man in the middle attacks: Analysis, motivation and prevention. *International Journal of Computer Networks and Communications Security*, 8:52–58.

Karie, N. M., Kebande, V. R., Ikuesan, R. A., Sookhak, M., and Venter, H. S. (2020). Hardening saml by integrating sso and multi-factor authentication (mfa) in the cloud. In *Proceedings of the 3rd International Conference on Networking, Information Systems and Security*, NISS2020, New York, NY, USA. Association for Computing Machinery.

Keshavarzi, A., Haghighat, A. T., and Bohlouli, M. (2020). Research challenges and prospective business impacts of cloud computing: A survey.

Lim, C. and Lukito, L. (2013). Malware attacks intelligence in higher education networks.

Mehraj, S. and Banday, M. T. (2020). Establishing a zero trust strategy in cloud computing environment. pages 1–6.

Mokoena, T. and Zuva, T. (2017). Malware analysis and detection in enterprise systems. In *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, pages 1304–1310.

Nagarjun, P. and Ahamad, S. S. (2020). Cross-site scripting research: A review. *International Journal of Advanced Computer Science and Applications*, 11(4).

Nongbri, I., Hadem, P., and Chettri, S. (2018). A survey on single sign-on. 6.

Parthasarathy, V. and Kumar, V. (2012). Cloud computing for business sustainability. *Asia Pacific Journal of Management Research and Innovations*, 8:461–474.

Roumani, Y. (2021). Patching zero-day vulnerabilities: an empirical analysis. *Journal of Cybersecurity*, 7(1). tyab023.

Shi, Y. (2018). Data security and privacy protection in public cloud. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4812–4819.

Singh, J. and Chaudhary, N. K. (2022). Oauth 2.0 : Architectural design augmentation for mitigation of common security vulnerabilities. *Journal of Information Security and Applications*, 65:103091.

Sophos (2020). The state of cloud security 2020.

Srivastava, P. and Khan, R. (2018). A review paper on cloud computing. *International Journal of Advanced Research in Computer Science and Software Engineering*, 8:17.

Sun, X. (2018). Critical security issues in cloud computing: A survey. In *2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)*, pages 216–221.

Sun, Y., Zhang, J., Xiong, Y., and Zhu, G. (2014). Data security and privacy in cloud computing. *International Journal of Distributed Sensor Networks*, 10(7):190903.

Surbiryala, J. and Rong, C. (2019). Cloud computing: History and overview. In *2019 IEEE Cloud Summit*, pages 1–7.

Tripathi, N. and Mehtre, B. (2013). Dos and ddos attacks: Impact, analysis and countermeasures. pages 1–6.

Vanitha, K., UMA, S. V., and Mahidhar, S. (2017). Distributed denial of service: Attack techniques and mitigation. In *2017 International Conference on Circuits, Controls, and Communications (CCUBE)*, pages 226–231.

Verma, A., Pedrosa, L., Korupolu, M. R., Oppenheimer, D., Tune, E., and Wilkes, J. (2015). Large-scale cluster management at Google with Borg. In *Proceedings of the European Conference on Computer Systems (EuroSys)*, Bordeaux, France.

Wong, D. (2021). *Real-World Cryptography*. Manning.