**CYBERYOUTH**

**Nonformal education for cyber-security training & resilience of youth organisations and young people**

***Cybersecurity online youth academy***

****

**Secure System Architecture and Design**

**Table of Contents**

[Introduction](#_heading=h.fpb3yt6tcffi)

[What you will learn](#_heading=h.bd1t10x6k115)

[Why it is important and how it can help you in everyday life](#_heading=h.1fqj6c1n3ygs)

[What career you can pursue](#_heading=h.l25siuf1d90q)

[Pre-requisites](#_heading=h.obikvf4haym5)

[Material](#_heading=h.mqxaktyycq90)

3[.1 Cloud Models](#_heading=h.ge2p7u5o14gx)

[Software as a Service(SaaS)](#_heading=h.rd7o0hytxuqd)

[Advantages of SaaS](#_heading=h.wlk220umah7p)

[Platform as a Service (PaaS)](#_heading=h.c095m7bxj2fq)

[Virtualization in Cloud Computing and Types](#_heading=h.n0jqiss3ax0a)

[Virtualization](#_heading=h.2edutmnfnzih)

[Work of Virtualization in Cloud Computing](#_heading=h.tk1v8iy64phx)

[Benefits of Virtualization](#_heading=h.yvuba4c2gfr)

[Characteristics of Virtualization](#_heading=h.pxq77x7hnppa)

[Types of Virtualization](#_heading=h.uwnxcibgsl5c)

[Uses of Virtualization](#_heading=h.b2mmjmsuxuc6)

Conclusions

3.2 Secure Application Development and Deployment

Secure Dev Ops

[How does DevSecOps work?](#_heading=h.yavonjy8xlkb)

[What are the advantages of DevSecOps?](#_heading=h.nhob7kzh9a69)

Provisioning and Deprovisioning

[Why user provisioning and deprovisioning matters](#_heading=h.bs2gvscqr4b2)

Secure Coding Techniques

[Why Is Secure Coding Important?](#_heading=h.sd4ye1ogf5qr)

Code Quality and Testing

[What is code quality?](#_heading=h.onyynaib9qua)

[What about security?](#_heading=h.t3de6m9suq19)

[OWASP](#_heading=h.2yn1dwymvsk1)

Conclusions

3.[3 Authentication Token Types](#_heading=h.40iaivf3ec0m)

[Authentication Methods](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

[Password Authentication Protocol (PAP),](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

[Authentication Token](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

[Symmetric-Key Authentication](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

[Biometric Authenticatio](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)n

[Authorization](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

[API Keys,](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

[Basic Auth](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

[HMAC](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

[OAuth](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

Conclusions

3.4 Cyber-Resilience Implementation

Redundancy

Backup Types

Best Techniques

Conclusions

3.5 Securing Embedded Systems

Embedded Systems

IoT

Conclusions

3.6 Basics Cryptographics Concepts

Digital Signatures,

Key Exchange,

Symmetric and Asymmetric

Transport encryption

Hashing

Steganography

Conclusions

[The Geek’s Corner](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.floxyjcfjlz3)

Conclusions

Quiz

References

# 

# 

# 

# 

# **Introduction**

## **What you will learn**

In this unit, you will learn various ways to understand how to secure and design an IT system to meet its security requirements, balancing this with its functional requirements. You will learn how security architecture works as a unified security design that addresses the necessities and potential risks involved in a certain scenario or environment, when and where to apply security controls and more.

## **Why it is important and how it can help youth organizations and youth in everyday life**

Secure system design and architecture is important because they solve complex security problems by selecting the best available solutions from a range of technological components and structures. The decisions are made fundamentally determine whether an organisation can manage its data, information systems and communication networks securely. Security architecture can translate each unique requirement into executable strategies and develop a risk-free environment for an organisation or business while aligning with the latest security standards and organisational/business needs.

Another importance is the principle of separation of privilege, which is often implemented by dividing a system into different levels, with each level having its own set of privileges. This principle is an important part of security design when designing any system as a core approach.

## **What career you can pursue**

Among the many career titles, some of the most important careers are;

-Security Engineer, who build security systems by implementing and monitoring

security controls with the aim of protecting an organisation’s data from cyber-attacks, loss or

unauthorised access.

To elaborate, here are the some tasks and responsibilities found on real security engineer

job listings on LinkedIn by Coursera.org:

* Identifying security measures to improve incident response
* Responding to security incidents
* Performing security assessments and code audits
* Developing technical solutions to security vulnerabilities
* Researching new attack vectors and developing threat models
* Automating security improvements

Also, there are other careers named as application security engineer, cloud engineer or

cloud architect, which are all related to security engineering main career field.

## **Pre-requisites**

To pursue a career in security field, here are the some core areas (pre-requised) and

degrees someone should have;

Degrees: Degree in Computer Science, IT, Systems Engineering or cybersecurity, or a

similar field.

Security engineers may also work in environments that require government-issued security

clearances or industry certifications such as:

* [Certified Information Systems Security Professional (CISSP)](https://cybersecurityguide.org/bootcamps/cissp-bootcamp/)
* [SANS](https://www.sans.org/apac/)/[GIAC](https://www.giac.org/) certification (various)
* [Certified Information Security Manager (CISM)](https://cybersecurityguide.org/bootcamps/cism-bootcamps/)
* [CompTIA Security+](https://cybersecurityguide.org/bootcamps/comptia-security-plus-bootcamp/)
* [Certified Information Systems Auditor (CISA)](https://cybersecurityguide.org/programs/cybersecurity-certifications/cisa/)

According to a study of current cybersecurity engineers [7], there are some foundational

skills and experiences in common including:

-An understanding of computer code, and in particular what dangerous code such as virus or

Malware looks like and hot to deal with it.

-Background and knowledge of risk assessment technologies and methods.

-Understanding of [computer forensics](https://cybersecurityguide.org/careers/computer-forensics/) and security breach protocols.

-The ability to perform security risk assessments and evaluations.

-Experience with developing and implementing security procedures and policies.

-Understanding of anti-virus software, firewall maintenance, and hacker detection.

Another skills requested for cybersecurity engineers as stated by [Cyberseek](https://www.cyberseek.org/pathway.html):

* Information security
* Network security
* Linux
* Information systems
* Python
* Cryptography
* Cisco
* Authentication

# 

# **Material**

# 

# **3.1 Cloud Models**

**Cloud Computing**

In its most simplest terms, cloud computing means storing and accessing the data

and programs on remote servers that are hosted on the internet, rather than the

computer’s hard drive or local server. Cloud computing is also referred to as

Internet-based computing.

**Cloud Computing Architecture:** Cloud computing architecture refers to the

components and sub-components required for cloud computing. These

components typically refer to:

1. Front end(fat client, thin client)

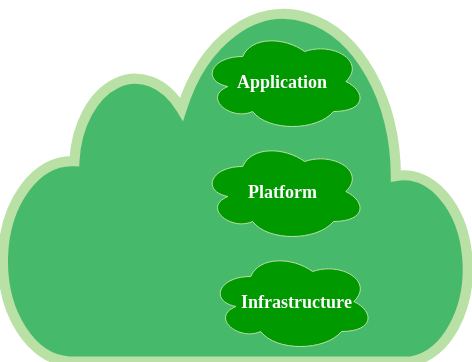
2. Back-end platforms(servers, storage)

3. Cloud-based delivery and a network (Internet, Intranet, Intercloud).

**Hosting a cloud:** There are three layers in cloud computing. Companies use these

layers based on the service they provide.

* Infrastructure
* Platform
* Application



THREE LAYERS OF CLOUD COMPUTING [4]

**What is Hosting?**

In the simplest term, hosting is the process of outsourcing an organization’s storage

resources to a service provider that offers its infrastructure services in a utility model [19].

**Benefits of Cloud Hosting**

1. **Scalability:** With Cloud hosting, it is easy to grow and shrink the number and

size of servers based on the need. This is done by either increasing or

decreasing the resources in the cloud. This ability to alter plans due to

fluctuation in business size and needs is a superb benefit of cloud computing,

especially when experiencing a sudden growth in demand [39].

**2. Instant:** Whatever you want is instantly available in the cloud.

**3. Save Money:** An advantage of cloud computing is the reduction in hardware

costs. Instead of purchasing in-house equipment, hardware needs are left to the

vendor. For entities that are growing rapidly, new hardware can be large,

expensive, and inconvenient. Cloud computing alleviates these issues because

resources can be acquired quickly and easily. Even better, the cost of repairing or

replacing equipment is passed to the vendors. Along with purchase costs, off-site

hardware cuts internal power costs and saves space. Large data centres can

take up precious office space and produce a large amount of heat. Moving to

cloud applications or storage can help maximise space and significantly cut

energy expenditures [28].

**4. Reliability:** Rather than being hosted on one single instance of a physical

Server, hosting is delivered on a virtual partition that draws its resource, such as

disk space, from an extensive network of underlying physical servers. If one

server goes offline it will have no effect on availability, as the virtual servers will

continue to pull resources from the remaining network of servers [52].

**5**. **Physical Security:** The underlying physical servers are still housed within data

centers and it helps to benefit from the security measures in a way that those

facilities implement to prevent people from accessing or disrupting them on-site[5]

**6**.**Outsource Management:** When you are managing your organisation someone

else manages your computing infrastructure. You do not need to worry about

management as well as upgradation.[8]

**Cloud Providers**

Cloud Computing can be defined as the practice of using a network of remote servers

hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer. Companies offering such kinds of [cloud computing](https://www.geeksforgeeks.org/architecture-of-cloud-computing/) services are called [cloud providers](https://www.geeksforgeeks.org/top-5-cloud-platform-service-providers-in-2020/) and typically charge for cloud computing services based on usage.Grids and

clusters are the foundations for cloud computing [9 and [www.geeksforgeeks.org](http://www.geeksforgeeks.org)]

Cloud computing has been effective in changing the commercial deployment of the systems

as some examples are:

1. **Amazon Web Services(AWS):** One of the most successful cloud-based businesses is

Amazon Web Services(AWS), which is an Infrastructure as a Service (Iaas) offering that

pays rent for virtual computers on Amazon’s infrastructure.

1. **Microsoft Azure Platform**: Microsoft is creating the Azure platform which enables

The .NET Framework Application to run over the internet as an alternative platform for

Microsoft developers. This is the classic Platform as a Service(PaaS).

**3. Google:** Google has built a worldwide network of data centres to service its search engi

ne. From this service, Google has captured the world’s advertising revenue. By using that

revenue, Google offers free software to users based on infrastructure. This is called

Software as a Service(SaaS).

**4. IBM Cloud** is a collection of cloud computing services for organisations provided by the

IBM Corporation. It provides infrastructure as a service, software as a service, and

platform as a service.

**5**. **Oracle Cloud** is a collection of cloud services offered by Oracle Corporation, including

infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service

(SaaS).

**6. Alibaba Cloud** is the cloud computing arm of Alibaba Group, providing a

comprehensive suite of global cloud computing services to power both their international

customers’ online businesses and Alibaba Group’s own e-commerce ecosystem.[38]

These are just a few examples of the wide range of applications for cloud computing. As

technology continues to advance, the possibilities for cloud computing will continue to

expand.

**Types of Cloud Computing**

Most cloud computing services fall into five broad categories:

1. Software as a service (SaaS)

2. Platform as a service (PaaS)

3. Infrastructure as a service (IaaS)

4. Anything/Everything as a service (XaaS)

5. Function as a Service (FaaS)

These are sometimes called the **cloud computing stack** because they are built on top of

one another. Knowing what they are and how they are different, makes it easier to choose

the most appropriate services for the users. These abstraction layers can also be viewed as

a **layered architecture** where services of a higher layer can be composed of services of the

underlying layer i.e, SaaS can provide Infrastructure. [22]

### **Software as a Service(SaaS)**

[Software-as-a-Service (SaaS)](https://www.geeksforgeeks.org/software-as-a-service-saas/) is a way of delivering services and applications over

the Internet. Instead of installing and maintaining software, you can simply access it

via the Internet, freeing yourselves from the complex software and hardware management

It removes the need to install and run applications on our own computers or in the data

centres eliminating the expenses of hardware as well as software maintenance.

SaaS provides a complete software solution that you purchase on a pay-as-you-go basis from a cloud service provider. Most SaaS applications can be run directly from a web browser without any downloads or installations required. The SaaS applications are sometimes called Web-based software, on-demand software, or hosted software.

#### 

#### 

#### 

#### 

#### 

#### 

#### **Advantages of SaaS**

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

1. **Cost-Effective:** Pay only for what you use.

**2. Reduced time:** Users can run most SaaS apps directly from their web browser

without needing to download and install any software. This reduces the time spent in

installation and configuration and can reduce the issues that can get in the way of

the software deployment.

**3. Accessibility:** We can Access app data from anywhere.

**4. Automatic updates:** Rather than purchasing new software, customers rely

on a SaaS provider to automatically perform the updates.

**5. Scalability:** It allows the users to access the services and features

on-demand.

Some of the various companies providing *Software as a service* are; Cloud9 Analytics,

Salesforce.com, Cloud Switch, Microsoft Office 365, Big Commerce, Eloqua, dropBox, and

Cloud Tran.

**Disadvantages of Saas :**

1. **Limited customization**: SaaS solutions are typically not as customizable as

on-premises software, meaning that users may have to work within the

constraints of the SaaS provider’s platform and may not be able to tailor the

software to their specific needs.

1. **Dependence on internet connectivity**: SaaS solutions are typically

cloud-based,which means that they require a stable internet connection to

function properly. This can be problematic for users in areas with poor

connectivity or for those who need to access the software in offline environments.

1. **Security concerns:** SaaS providers are responsible for maintaining the

security of the data stored on their servers, but there is still a risk of data

breaches or other security incidents.

1. **Limited control over data:** SaaS providers may have access to a user’s data

which can be a concern for organizations that need to maintain strict control over

their data for regulatory or other reasons.

### 

### **Platform as a Service (PaaS)**

[PaaS](https://www.geeksforgeeks.org/platform-as-a-service-paas-and-its-types/) is a category of cloud computing that provides a platform and environment to allow

developers to build applications and services over the internet. PaaS services are hosted in

the cloud and accessed by users simply via their web browser

A PaaS provider hosts the hardware and software on its own infrastructure. As a result, PaaS

frees users from having to install in-house hardware and software to develop or run a new

application. Thus, the development and deployment of the application take place

independent of the hardware.

The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment. To make it simple, take the example of an annual day function, you will have two options either to create a venue or to rent a venue but the function is the same.[33]

**Advantages of PaaS:**

**1**.Simple and convenient for users: It provides much of the infrastructure and other IT

services, which users can access anywhere via a web browser.

**2**.Cost-Effective: It charges for the services provided on a per-use basis thus eliminating the

expenses one may have for on-premises hardware and software.

**3**. Efficiently managing the lifecycle: It is designed to support the complete web application

lifecycle: building, testing, deploying, managing, and updating.

**4**. Efficiency: It allows for higher-level programming with reduced complexity thus, the

overall development of the application can be more effective.

The various companies providing Platform as a service are Amazon Web services Elastic,

Beanstalk, Salesforce, Windows Azure, Google App Engine, cloud Bees and IBM smart cloud. [49]

**Disadvantages of Paas:**

**1.** Limited control over infrastructure: PaaS providers typically manage the underlying

infrastructure and take care of maintenance and updates, but this can also mean that users

have less control over the environment and may not be able to make certain customizations.

**2.**Dependence on the provider: Users are dependent on the PaaS provider for the

availability, scalability, and reliability of the platform, which can be a risk if the provider

experiences outages or other issues.[38]

**3.**Limited flexibility: PaaS solutions may not be able to accommodate certain types of

workloads or applications, which can limit the value of the solution for certain

organizations.

There are other services such as ‘’ Infrastructure as a Service’’ (IaaS),Hardware as a

Service (HaaS), Function as a Service (FaaS) which can be learnt from various

resources. (not included in this document)

# 

# **Virtualization in Cloud Computing and Types**

# **Virtualization**

Virtualization is a technique concerning how to separate a service from the

underlying physical delivery of that service. It is the process of creating a virtual

version of something like computer hardware. It was initially developed during the

mainframe era. It involves using specialized software to create a virtual or

software-created version of a computing resource rather than the actual version of

the same resource. With the help of Virtualization, multiple operating systems and

applications can run on the same machine and its same hardware at the same time,

increasing the utilization and flexibility of hardware. [3]

In other words, one of the main cost-effective, hardware-reducing, and

energy-saving techniques used by cloud providers is Virtualization. Virtualization

allows sharing of a single physical instance of a resource or an application among

multiple customers and organizations at one time. It does this by assigning a logical

name to physical storage and providing a pointer to that physical resource on

demand. The term virtualization is often synonymous with hardware virtualization

which plays a fundamental role in efficiently delivering Infrastructure-as-a-Service

(IaaS) solutions for [cloud computing](https://www.geeksforgeeks.org/cloud-computing/). Moreover, virtualization technologies provide a

virtual environment for not only executing applications but also for storage, memory,

and networking.[6]

## **Work of Virtualization in Cloud Computing**

Virtualization has a prominent impact on Cloud Computing. In the case of cloud

computing, users store data in the cloud, but with the help of Virtualization, users

have the extra benefit of sharing the infrastructure. Cloud Vendors take care of the

required physical resources, but these cloud providers charge a huge amount for

these services which impacts every user or organization. Virtualization helps Users

or Organisations in maintaining those services which are required by a company

through external (third-party) people, which helps in reducing costs to the company.

This is the way through which Virtualization works in Cloud Computing. [6, 13]

## **Benefits of Virtualization**

* More flexible and efficient allocation of resources.
* Enhance development productivity.
* It lowers the cost of IT infrastructure.
* Remote access and rapid scalability
* High availability and disaster recovery.
* Pay per use of the IT infrastructure on demand.
* Enables running multiple operating systems.[2,17]

## **Characteristics of Virtualization**

* **Increased Security:** The ability to control the execution of a guest program in a

completely transparent manner opens new possibilities for delivering a secure,

controlled execution environment. All the operations of the guest programs are

generally performed against the virtual machine, which then translates and

applies them to the host programs. [8, 38]

* **Managed Execution:** In particular, sharing, aggregation, emulation, and

isolation are the most relevant features.

* **Sharing:** Virtualization allows the creation of a separate computing

environment within the same host.

## **Types of Virtualization**

1. **Application Virtualization**

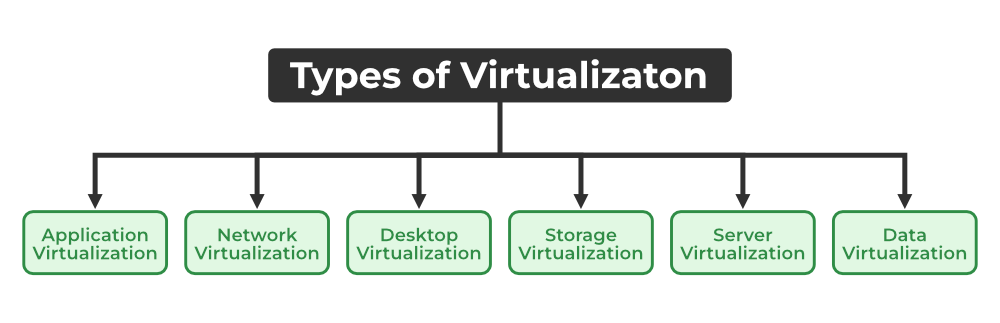
**2.** [**Network Virtualization**](https://www.geeksforgeeks.org/network-virtualization-in-cloud-computing/)

**3. Desktop Virtualization**

**4. Storage Virtualization**

**5.** [**Server Virtualization**](https://www.geeksforgeeks.org/server-virtualization/)

**6. Data virtualization**

****

**Source:** [**www.techradar.com**](http://www.techradar.com/)

1. Application Virtualization: Application virtualization helps a user to have

remote access to an application from a server. The server stores all personal

information and other characteristics of the application but can still run on a

local workstation through the internet. An example of this would be a user who

needs to run two different versions of the same software. Technologies that

use application virtualization are hosted applications and packaged

Applications.[5, 27]

2. Network Virtualization: The ability to run multiple virtual networks with each

having a separate control and data plan. It co-exists together on top of one physical

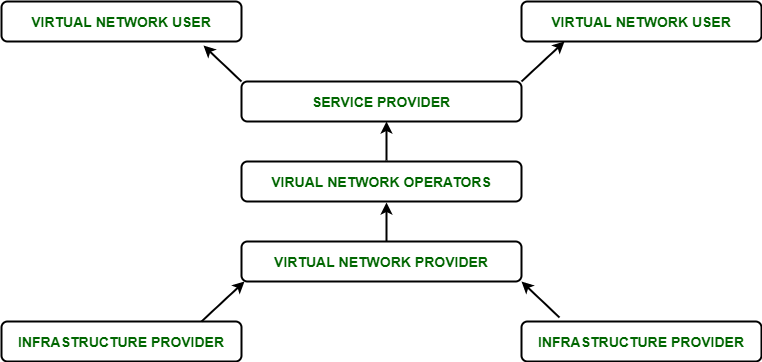
network. It can be managed by individual parties that are potentially confidential to

each other. Network virtualization provides a facility to create and provision virtual

networks, logical switches, routers, [firewalls](https://www.geeksforgeeks.org/introduction-of-firewall-in-computer-network/), load balancers, [Virtual Private](https://www.geeksforgeeks.org/virtual-private-network-vpn-introduction/)

[Networks (VPN)](https://www.geeksforgeeks.org/virtual-private-network-vpn-introduction/), and workload security within days or even weeks.

(www.geeksforgeek.org)[9]



Source: www.Cloudtech.com

use application virtualization are hosted applications and packaged

applications.

3.Desktop Virtualization: Desktop virtualization allows the users’ OS to be remotely

stored on a server in the data center. It allows the user to access their desktop

virtually, from any location by a different machine. Users who want specific operating

systems other than Windows Server will need to have a virtual desktop. The main

benefits of desktop virtualization are user mobility, portability, and easy management

of software installation, updates and patches. [11, 31]

4. Storage Virtualization: Storage virtualization is an array of servers that are

managed by a virtual storage system. The servers aren’t aware of exactly where

their data is stored and instead function more like worker bees in a hive. It makes

managing storage from multiple sources be managed and utilized as a single

repository. storage virtualization software maintains smooth operations, consistent

performance, and a continuous suite of advanced functions despite changes, breaks

down, and differences in the underlying equipment.[14, 42]

5. Server Virtualization: This is a kind of virtualization in which the masking of

server resources takes place. Here, the central server (physical server) is divided

into multiple different virtual servers by changing the identity number, and

processors. So, each system can operate its operating systems in an isolated

manner. Where each sub-server knows the identity of the central server. It causes an

increase in performance and reduces the operating cost by the deployment of main

server resources into a sub-server resource. It’s beneficial in virtual migration,

reducing energy consumption, reducing infrastructural costs, etc.[19, 48]

6. Data Virtualization: This is the kind of virtualization in which the data is collected from

various sources and managed at a single place without knowing more about the technical

information like how data is collected, stored & formatted then arranged that datalogically

so that its virtual view can be accessed by its interested people and stakehold

ers, and users through the various cloud services remotely. Many big giant companies are

providing their services like Oracle, IBM, At scale, Cdata, etc.[12]

## **Uses of Virtualization**

* Data-integration
* Business-integration
* Service-oriented architecture data-services
* Searching organizational data
* Searching organizational data

**Conclusions**

Cloud computing provides great opportunities for organisations\businesses\people as listed in this part, which has been gaining attention from public and private industries since the last 10 years, however, it can be said that it is still not properly recognised\learned well, in that, it has a long way to go based on the online storage need of data in the World. In addition, looking at the benefits of cloud computing which can provide, it is assumed that it will be widely used by people and organisations in near future and relative investments will be made from the industries. This can be predicted by the number of cloud providers out there, who provide cloud services as a service, its platforms and tools etc.

**3**[**.2**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.wv8lmqsplpdd) **Secure Application Development and Deployment**

**Secure DevOps**

DevOps is a set of techniques that stresses collaboration and automation between

development and operations teams. DevSecOps is the integration of security

practices into this methodology. DevSecOps seeks to establish a security culture that

guarantees the software is secure and complies with compliance standards by

integrating security into every phase of the software development lifecycle, from

planning through deployment.[39, 47]

### 

### 

### 

### 

### 

### **How does DevSecOps work?**

What does DevSecOps look like in practice? DevSecOps is fundamentally about cooperation and communication between teams working on development, security, and operations. This implies that everyone bears some kind of responsibility for security, not just the security team. Instead of adding security features later, development teams collaborate with security and operations teams to incorporate security into the software from the start.

The automation of DevSecOps is a crucial element. Automation makes the development

process more efficient, reduces errors, and ensures consistency. DevSecOps can aid in the

quicker and more precise detection of vulnerabilities and threats by automating security

testing and other security operations.

### **What are the advantages of DevSecOps?**

The ability to identify and address security risks earlier in the development process is one of

the main advantages of DevSecOps. This means that security is incorporated into the

software at the outset instead of being added later, which can be expensive and

time-consuming. Also, DevSecOps plays a big role in decreased risk of security breaches and data leaks by identifying vulnerabilities earlier.

The fact that DevSecOps helps to ensure compliance with laws and standards is another

crucial feature of the practice. In many businesses, especially those that deal with sensitive or private data, including healthcare and banking, compliance is becoming more and more

crucial. DevSecOps aids in ensuring that the software complies with requirements by

incorporating compliance into the development process.[50]

**Provisioning and Deprevisioning**

**What Is Provisioning and Deprovisioning?**

Provisioning is the process of making information technology (IT) systems available to

users. Depending on your organization’s needs, provisioning can be defined at the

network, server, application, and user level:

* Network provisioning involves setting up a network that can be accessed by users, servers, and devices. The telecommunications industry, for example, uses network provisioning to provide customers with wireless solutions.
* Server provisioning is the process of setting up a server that can be used within a network. This may include creating a new machine, putting physical hardware in a data center, installing and configuring software, and connecting to networks and storage.
* Application provisioning is an infrastructure management solution that allows administrators to optimize performance for various environments within an enterprise.
* User provisioning is the process of managing digital identities, which includes creating, updating, and removing rights and permissions to a business’s applications, files, networks, systems, and resources.

Deprovisioning is the process of removing user access to software and network services. Put simply, it’s the exact opposite of provisioning—and typically occurs when employees change roles or leave a company.

Both provisioning and deprovisioning play an important role in securing IT systems and applications, but effective and automated user provisioning should be top of mind for any organization that wants to enhance their security posture.

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## **Why user provisioning and deprovisioning matters**

When a new employee is hired, one of the first things an organization does is create a record of that employee. It then becomes the responsibility of HR, IT, or a combination of the two teams, to provide that employee with access to all of the apps, accounts, and systems they need to do their job.

User provisioning, therefore, takes place whenever information is added or amended in your organization’s HR systems: this includes the addition of team members, role changes, promotions, and department transfers, among other things. In other words, user provisioning helps you provide the right level of access to the right users during onboarding, update access throughout employment, and—during the deprovision process—remove access when an employee leaves the organization.

**What is automated provisioning and what are its benefits?**

Automated provisioning means making the manual processes of onboarding and offboarding users automatic. In organizations both big and small, automated user provisioning frees up IT and HR to work on more strategic tasks, prevents gaps in security by minimizing the impact of human error, and provides better user experiences.

Manually updating individual user profiles, account privileges, and group memberships requires time, especially as employees need access to more workplace applications than ever. Not only that, but the process can get delayed if IT teams are busy working on other projects. This can prevent new users from being onboarded quickly, stall the grant or removal of access rights, and make it harder to monitor and identify irrelevant permissions. Automated provisioning and deprovisioning takes this pressure off IT’s shoulders, allowing everyone to spend time on projects that drive business value.[48]

By automating user provisioning, you can also eliminate some of the gaps in access management that could leave your business vulnerable to security breaches. Manually creating user accounts means that someone within an organization must share a password with an employee—which is likely a very insecure process, such as sending an email or writing on a sticky note. Similar instances of human error may also pose a threat to security. For example, users could accidentally be provisioned to systems and data that they shouldn’t have access to, or still have access once they leave your organization.

Automating user provisioning and deprovisioning removes these risks, providing individuals with permissions in a safe and private manner. The process ensures that a user is provisioned for on-premises and external apps based on their role’s attributes. These attributes and permissions are then stored in one central location, ensuring they can be easily modified as an employee’s role changes. When departments or teams implement a new tool or modify employees’ entitlements, access can also be rolled out based on group rules.[30]

Provisioning IT automatically provides users with access only when it is necessary, preventing any security gaps that hackers could exploit to gain unauthorized access to sensitive corporate information.

**Secure Coding Techniques**

**What is secure coding?**

Secure coding, also referred to as secure programming, involves writing code in a high-level language that follows strict principles, with the goal of preventing potential vulnerabilities (which could expose data or cause harm within a targeted system).

Secure coding is more than just writing, compiling, and releasing code into applications. To fully embrace secure programming, you also need to create a secure development environment built on a reliable and secure IT infrastructure using secure hardware, software, and services and providers.[23, 43]

## [**Why Is Secure Coding Important?**](https://snyk.io/learn/secure-coding-practices/#important)

More and more financial transactions are also moving online. Security incidents often originate deep in an application’s underlying software and can have serious consequences for businesses and individuals alike. Insecure code in important industries (e.g., finance, healthcare, energy, and transport) could result in financial and property damages, market manipulation and theft, even physical harm and fatalities.

And the danger is real: Media reports in recent years have highlighted just how insecure much of the software we use is. Even major organizations with the resources and knowledge at their disposal have experienced serious data breaches. For companies that provide software to consumers or enterprises, customer trust is of course extremely valuable, and losing that trust could impact their bottom line. Ensuring secure coding practices therefore must be a top priority for these organizations. [16, 41]

[**5 Best practices for secure coding**](https://snyk.io/learn/secure-coding-practices/#best)**:**

Code minification and obfuscation: Making your code harder to access, and by extension harder to read, can deter potential attackers. In the JavaScript world, a common practice is to minify code. Minification removes white space and line breaks from your code. And while it is and is really intended to enhance performance by reducing the footprint of code files, it has the added benefit of making exposed code much harder to read. Another similar, more effective technique is code obfuscation, which turns human-readable code into text that is difficult to understand.[51]

Avoiding shortcuts: It can be tempting for developers to want to take shortcuts to release code into production faster, but this could have serious security implications. For example, attacks often occur when hardcoded credentials and security tokens are left as comments. This information should be cleaned up long before your apps are released. But as your code base gets bigger and there is mounting pressure to deliver working code on increasingly tight release schedules, the likelihood of security gaps goes up.

Automated scanning & code reviews: Cross-site scripting (XSS), SQL injection, and other types of attacks can exploit security vulnerabilities in your code. Both XSS and SQL injection attacks result from weakness in your code that fails to distinguish between data and commands. XSS executes malicious code under your domain. SQL injection attacks attempt to steal or manipulate data in your internal data stores. A combination of regular secure code reviews and automated tools that scan your code for these vulnerabilities can help prevent such attacks.

Avoiding components with known vulnerabilities: While open-source components and libraries, often consumed as packages, can save developers time and energy, they are also a common entry point for malicious actors and a great source of vulnerabilities and potential exploits. Refraining from using those components with known vulnerabilities and constantly monitoring for new vulnerabilities throughout the development process in the components you use will help you maintain the integrity of your code.[30, 47]

Auditing & logging: Software with sufficient logging and monitoring will allow you to detect potential incidents when your code is deployed in a production environment.

**Version Control and Change Management**

**What is Version Control**

Version control, also known as source control, is the practice of tracking and managing changes to software code. Version control systems are software tools that help software teams manage changes to source code over time. As development environments have accelerated, version control systems help software teams work faster and smarter. They are especially useful for DevOps teams since they help them to reduce development time and increase successful deployments.[24, 33]

Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

For almost all software projects, the source code is like the crown jewels - a precious asset whose value must be protected. For most software teams, the source code is a repository of the invaluable knowledge and understanding about the problem domain that the developers have collected and refined through careful effort. Version control protects source code from both catastrophe and the casual degradation of human error and unintended consequences.[29, 30]

Software developers working in teams are continually writing new source code and changing existing source code. The code for a project, app or software component is typically organized in a folder structure or "file tree". One developer on the team may be working on a new feature while another developer fixes an unrelated bug by changing code, each developer may make their changes in several parts of the file tree.[10, 39]

Version control helps teams solve these kinds of problems, tracking every individual change by each contributor and helping prevent concurrent work from conflicting. Changes made in one part of the software can be incompatible with those made by another developer working at the same time. This problem should be discovered and solved in an orderly manner without blocking the work of the rest of the team. Further, in all software development, any change can introduce new bugs on its own and new software can't be trusted until it's tested. So testing and development proceed together until a new version is ready. [10, 31]

**What is Change Management?**

IT change management is a structured process for evaluating proposed IT system or service changes. This procedure is carried out prior to implementing the requested change on an organization’s network, reducing or eliminating network outages.

Information Technology (IT) change management is a structured process for reviewing proposed IT system or service changes. This process occurs prior to implementing the requested change on an organization’s network, thus minimizing or eliminating network outages.

IT change management is necessary to ensure any changes to the network will not degrade the performance of the network. Any changes to the network should be a defined, purposeful action to eliminate a found vulnerability, upgrade a component on the network for improved performance, or replace a currently obsolete or faulty network component.[32]

**What Are the Types of IT Changes?**

IT system or service changes are categorized into three types:

**Standard Changes**

Standard changes are routine, and follow a pre-established process regarding

risk analysis and pre-approvals. These changes are vetted processes that

have been pre-approved for execution. Examples of standard changes include

the following:

Upgrading RAM or hard drive size

Replacing a failing network device

Making a new database instance [45]

**Normal Changes**

Normal changes do not have a pre-established process. A risk analysis and

deployment plan must be submitted for approval prior to implementing these

changes on the IT network. Examples of normal changes include the following:

Upgrading to a new [compliance management system](https://www.cioinsight.com/blogs/dont-overlook-it-risk-compliance-when-defending-against-cyberattacks/)

Upgrading network devices for improved performance

Relocating a server farm [45]

**Emergency Changes**

Emergency changes are when an unplanned outage has occurred, or is likely to

occur, due to a discovered vulnerability that possesses a significant threat to the

network. Examples of emergency changes are the following:

Installing a security patch

A network device outage

Recovering from a major incident (i.e. fiber strand cut) [45, 48]

**Code Quality and Security**

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## **What is code quality?**

When we look up the word "quality" [in the dictionary](https://www.oxfordlearnersdictionaries.com/us/definition/english/quality_1?q=quality), it is defined as "how good or bad something is." Therefore, when we talk about "code quality," we refer to a metric to qualify how good or bad a set of software instructions is. But how good or bad by virtue of what? Generally, on the basis of a variety of characteristics or attributes. However, as can happen, for instance, with the classification of human behavior from a moral perspective, code quality is not a definitive metric with always shared and objective parameters and is open to discussion. It depends on subjectivity, on what industries and organizations define based on their specific needs, requirements and approaches. Code quality, for example, will not necessarily be viewed in the same way by those who build simple mobile games and those who develop programs that control the machinery of enormous electric companies, both software with such distinct criticality.[31]

Even so, some specific ideas seem to be shared for almost all software projects that allow the separation of good and bad quality code. Bad or poor-quality code may lack coherence and consistency in handling conventions and be full of bugs and complexity. Good quality code, conversely, is usually seen as straightforward, bug-free, well-documented and fulfilling its intended function for its end users. The following are some of those key properties shared to qualify code quality:

• Reliability: It measures how likely it is that the software will work without failure,

accomplishing its purposes during a specific period. This property depends on the

number of errors present in the code.

• Robustness: It measures how well the software can cope with strange user behavior and other conditions using understandable error messages. This property is related to the reduced susceptibility to hidden bugs or the introduction of new bugs.

• Testability: It measures how well the software supports the use of tests that can be employed to, for example, verify certain behaviors or detect failures in its functionality.

• Readability: It measures how legible and understandable the software code is not only for its original authors but also for those who intend to review and edit it. It depends on the use of comments, notations, indentation, documentation and simplicity, among other things.

• Maintainability: It measures the ease with which the software can be maintained, i.e., repaired, updated and improved. This property depends on the code's structure, size, consistency and complexity.

• Portability: It measures how usable the software is on different devices, platforms or other environments. In other words, it measures how easily it can be transferred from one to another, depending on the number of required modifications.

• Reusability: It measures how much the software's pieces of code or assets can be replicated or reused (even to build upon them) by developers in other projects or programs.

Up to this point, what is problematic is that after roaming the web and glancing through various sources, we generally don't find security or at least not explicitly suggested among

these key properties.

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## **What about security?**

Code quality is usually associated with the code's performance and the experience of the end users and the developers that work on it. However, shouldn't code quality also be measured by its security? On the other hand, doesn't security also rely on what we call quality? Almost a decade ago, a group of researchers shared the following words in a study for the Software Engineering Institute:

Many of the Common Weakness Enumerations (CWEs), such as the improper use of programming language constructs, buffer overflows, and failures to validate input values, can be associated with poor quality coding and development practices. Improving quality is a necessary condition for addressing some software security issues.[11]

Poor-quality pieces of code, even minor errors, resulting from inadequate coding practices, give rise to security weaknesses and vulnerabilities that can be exploited by malicious hackers and generate substantial negative impacts on organizations and end users. Security thus depends on the quality of the product. But this is a different quality that involves not only properties such as those listed above. In this "upgraded code quality," standards and practices are introduced, which, if not complied with, lead us to speak of insecurity, just as non-compliance with other requirements could lead us to talk about, for instance, unreliability or low readability. By introducing these new standards, we have included security as a property in the previous list. Thus, henceforth, in our discourse, code security will be taken as a factor that plays a role in determining code quality.[11, 41]

Security vulnerabilities in the code jeopardize not only the valuable performance of the software but also the privacy of sensitive data in case they're processed there. The non-inclusion of security in the frequently seen code quality concept may be partly related to the latter. In the past, unlike nowadays, software products that did not involve the use of sensitive data of the organization and its users and customers or contact with external threats were more abundant. At that time, there was a more significant concern for properties such as reliability, testability and maintainability. Today, however, in our economy, there is much more dependence on the Internet and web and mobile applications that transfer sensitive data, as well as a greater number of IoT devices and interconnections. This is why it's increasingly necessary that security also appears as a key property in relation to code quality.[11, 41, 43]

**OWASP**

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## **What is OWASP?**

The Open Web Application Security Project, or OWASP, is an international non-profit organization dedicated to [web application security](https://www.cloudflare.com/learning/security/what-is-web-application-security/). One of OWASP’s core principles is that all of their materials be freely available and easily accessible on their website, making it possible for anyone to improve their own web application security. The materials they offer include documentation, tools, videos, and forums. Perhaps their best-known project is the OWASP Top 10. [7, 30]

**What is the OWASP Top 10?**

The OWASP Top 10 is a regularly-updated report outlining security concerns for web application security, focusing on the 10 most critical risks. The report is put together by a team of security experts from all over the world. OWASP refers to the Top 10 as an ‘awareness document’ and they recommend that all companies incorporate the report into their processes in order to minimize and/or mitigate security risks.

Below are the security risks reported in the OWASP Top 10 report:

**1.Injection**

Injection attacks happen when untrusted data is sent to a code interpreter through a form input or some other data submission to a web application. For example, an attacker could enter SQL database code into a form that expects a plaintext username. If that form input is not properly secured, this would result in that SQL code being executed. This is known as an SQL injection attack.

Injection attacks can be prevented by validating and/or sanitizing user-submitted data. (Validation means rejecting suspicious-looking data, while sanitization refers to cleaning up the suspicious-looking parts of the data.) In addition, a database admin can set controls to minimize the amount of information an injection attack can expose.

**2. Broken Authentication**

Vulnerabilities in authentication (login) systems can give attackers access to user accounts and even the ability to compromise an entire system using an admin account. For example, an attacker can take a list containing thousands of known username/password combinations obtained during a data breach and use a script to try all those combinations on a login system to see if there are any that work.

Some strategies to mitigate authentication vulnerabilities are requiring two-factor authentication (2FA) as well as limiting or delaying repeated login attempts using rate limiting.

**3. Sensitive Data Exposure**

If web applications don’t protect sensitive data such as financial information and passwords, attackers can gain access to that data and sellor utilize it for nefarious purposes. One popular method for stealing sensitive information is using an on-path attack.

Data exposure risk can be minimized by encrypting all sensitive data as well as disabling the caching\* of any sensitive information. Additionally, web application developers should take care to ensure that they are not unnecessarily storing any sensitive data.

\*Caching is the practice of temporarily storing data for re-use. For example, web browsers will often cache webpages so that if a user revisits thosepages within a fixed time span, the browser does not have to fetch the pages from the web.

**4. XML External Entities (XEE)**

This is an attack against a web application that parses XML\* input. This input can reference an external entity, attempting to exploit a vulnerability in the parser. An ‘external entity’ in this context refers to a storage unit, such as a hard drive. An XML parser can be duped into sending data to an unauthorized external entity, which can pass sensitive data directly to an attacker. [41]

The best ways to prevent XEE attacks are to have web applications accept a less complex type of data, such as JSON\*\*, or at the very least to patch XML parsers and disable the use of external entities in an XML application.

\*XML or Extensible Markup Language is a markup language intended to be both human-readable and machine-readable. Due to its complexity and security vulnerabilities, it is now being phased out of use in many web applications.

\*\*JavaScript Object Notation (JSON) is a type of simple, human-readable notation often used to transmit data over the internet. Although it was originally created for JavaScript, JSON is language-agnostic and can be interpreted by many different programming languages.

**5. Broken Access Control**

Access control refers a system that controls access to information or functionality. Broken access controls allow attackers to bypass authorization and perform tasks as though they were privileged users such as administrators. For example a web application could allow a user to change which account they are logged in as simply by changing part of a url, without any other verification.

Access controls can be secured by ensuring that a web application uses authorization tokens\* and sets tight controls on them.

\*Many services issue authorization tokens when users log in. Every privileged request that a user makes will require that the authorization token be present. This is a secure way to ensure that the user is who they say they are, without having to constantly enter their login credentials. [33]

**6. Security Misconfiguration**

Security misconfiguration is the most common vulnerability on the list, and is often the result of using default configurations or displaying excessively verbose errors. For instance, an application could show a user overly-descriptive errors which may reveal vulnerabilities in the application. This can be mitigated by removing any unused features in the code and ensuring that error messages are more general.[18, 23]

**7. Cross-Site Scripting**

Cross-site scripting vulnerabilities occur when web applications allow users to add custom code into a url path or onto a website that will be seen by other users. This vulnerability can be exploited to run malicious JavaScript code on a victim’s browser. For example, an attacker could send an email to a victim that appears to be from a trusted bank, with a link to that bank’s website. This link could have some malicious JavaScript code tagged onto the end of the url. If the bank’s site is not properly protected against cross-site scripting, then that malicious code will be run in the victim’s web browser when they click on the link.

Mitigation strategies for cross-site scripting include escaping untrusted HTTP requests as well as validating and/or sanitizing user-generated content. Using modern web development frameworks like ReactJS and Ruby on Rails also provides some built-in cross-site scripting protection.[25, 45]

**8. Insecure Deserialization**

This threat targets the many web applications which frequently serialize and deserialize data. Serialization means taking objects from the application code and converting them into a format that can be used for another purpose, such as storing the data to disk or streaming it. Deserialization is just the opposite: converting serialized data back into objects the application can use. Serialization is sort of like packing furniture away into boxes before a move, and deserialization is like unpacking the boxes and assembling the furniture after the move. An insecure deserialization attack is like having the movers tamper with the contents of the boxes before they are unpacked.

An insecure deserialization exploit is the result of deserializing data from untrusted sources, and can result in serious consequences like DDoS attacks and remote code execution attacks. While steps can be taken to try and catch attackers, such as monitoring deserialization and implementing type checks, the only sure way to protect against insecure deserialization attacks is to prohibit the deserialization of data from untrusted sources.[44]

**9. Using Components With Known Vulnerabilities**

Many modern web developers use components such as libraries and frameworks in their web applications. These components are pieces of software that help developers avoid redundant work and provide needed functionality; common example include front-end frameworks like React and smaller libraries that used to add share icons or a/b testing. Some attackers look for vulnerabilities in these components which they can then use to orchestrate attacks. Some of the more popular components are used on hundreds of thousands of websites; an attacker finding a security hole in one of these components could leave hundreds of thousands of sites vulnerable to exploit.[13, 47]

Component developers often offer security patches and updates to plug up known vulnerabilities, but web application developers don’t always have the patched or most-recent versions of components running on their applications. To minimize the risk of running components with known vulnerabilities, developers should remove unused components from their projects, as well as ensuring that they are receiving components from a trusted source and ensuring they are up to date.[13]

**10. Insufficient Logging And Monitoring**

Many web applications are not taking enough steps to detect data breaches. The average discovery time for a breach is around 200 days after it has happened. This gives attackers a lot of time to cause damage before there is any response. OWASP recommends that web developers should implement logging and monitoring as well as incident response plans to ensure that they are made aware of attacks on their applications.[49]

**Conclusions**

Secure application development directly addresses integrating security into every phase of the software development lifecycle, from planning through deployment. This is a highly needed operational area which works for establishing a security culture that guarantees the software is secure and complies with compliance standards. Similarly, automated provisioning means making the manual processes of onboarding and offboarding users automatic. In organizations both big and small, automated user provisioning frees up IT and HR to work on more strategic tasks, prevents gaps in security by minimizing the impact of human error, and provides better user experiences. Also, secure coding, also referred to as secure programming, involves writing code in a high-level language that follows strict principles, with the goal of preventing potential vulnerabilities which could expose data or cause harm within a targeted system.

**3**[**.3 Authentication and Authorization**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

[**Authentication Methods**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

[**Password Authentication Protocol (PAP)**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

PAP, or password authentication protocol, is a point-to-point protocol (PPP) authentication

method that uses passwords to validate users. It is an internet standard (RFC 1334),

password-based authentication protocol.

Using PAP, data is not encrypted. It is sent to the authentication server as plain text. PAP

uses a two-way handshake to authenticate users based on their provided username and

password.

When used in PPP, the password authentication protocol is considered a weak authentication scheme. Since data is unencrypted, it is vulnerable and visible to a bad actor who is able to view the PPP session.

Using CHAP (challenge-handshake authentication protocol) can add an extra layer of security to the PPP session by adding a three-way handshake process. PAP is a standard login procedure used as a PPP method for authenticating users. [6, 30]

[**Authentication Token**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

Token-based authentication is a protocol which allows users to verify their identity, and in return receive a unique access token. During the life of the token, users then access the website or app that the token has been issued for, rather than having to re-enter credentials each time they go back to the same webpage, app, or any resource protected with that same token.

Auth tokens work like a stamped ticket. The user retains access as long as the token remains valid. Once the user logs out or quits an app, the token is invalidated.

Token-based authentication is different from traditional password-based or server-based authentication techniques. Tokens offer a second layer of security, and administrators have detailed control over each action and transaction.

But using tokens requires a bit of coding know-how. Most developers pick up the techniques quickly, but there is a learning curve. [1, 40]

## 

## 

## 

## 

## 

## 

## 

## **3 Authentication Token Types**

All authentication tokens allow access, but each type works a little differently.

These are three common types of authentication tokens:

• Connected: Keys, discs, drives, and other physical items plug into the system for access. If you've ever used a USB device or smartcard to log into a system, you've used a connected token.

• Contactless: A device is close enough to a server to communicate with it, but it doesn't plug in. Microsoft's so-called "magic ring" would be an example of this type of token.

• Disconnected: A device can communicate with the server across long distances, even if it never touches another device at all. If you've ever used your phone for a two-factor authentication process, you've used this type of token.

In all three of these scenarios, a user must do something to start the process. They may need to enter a password or answer a question. But even when they complete those preliminary steps perfectly, they can't gain access without the help of an access token. [3, 22]

## **Token Authentication in 4 Easy Steps**

Use a token-based authentication system, and visitors will verify credentials just once. In return, they'll get a token that allows access for a time period you define.

The process works like this:

• Request: The person asks for access to a server or protected resource. That could involve a login with a password, or it could involve some other process you specify.

• Verification: The server determines that the person should have access. That could involve checking the password against the username, or it could involve another process you specify.

• Tokens: The server communicates with the authentication device, like a ring, key, phone, or similar device. After verification, the server issues a token and passes it to the user.

• Storage: The token sits within the user's browser while work continues.[8, 11]

If the user attempts to visit a different part of the server, the token communicates with the server again. Access is granted or denied based on the token.

Administrators set limits on tokens. You could allow a one-use token that is immediately destroyed when the person logs out. Or you could set the token to self-destruct at the end of a specified time period.

[**Symmetric-Key Authentication**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

A symmetric key is one that can be used both to encrypt and decrypt data. This means that to decrypt data, one should have the similar key that was used to encrypt it. Symmetric encryption is usually more effective than asymmetric encryption and therefore preferred when high amounts of data need to be exchanged.

It can be creating the shared key is complex using only symmetric encryption algorithms, so in some cases, an asymmetric encryption is used to create the shared key between two parties.[37]

The Digital Encryption Standard (DES) algorithm is a symmetric key cryptography approach commonly used smart card systems. This approach need a stored, secret cryptographic key and the public DES algorithm in each smart card and CAD. There are various steps in symmetric key authentication are as follows;

• The smart card sends the microprocessor serial number (I) to the CAD, which connect the number with the master key (Mk) to form the smart card's diversified key (K). The issuer loads a diversified key into every smart card during card initialization.

• The CAD makes a random number (R), thus encrypts R to form the value Y that is sent as the challenge to the smart card.

• The smart card decrypts Y, forming the response (X) and restore X to the CAD.

• The CAD compares R and X, accepting the card if the two values connect.[43]

The Telepass 1 algorithm is an algorithm used for smart card authentication. This algorithm needs a diversified secret key, the contents of a definite word in the smart card memory, and a random outside value to compute the response to an authentication challenge. The Telepass 1 algorithm supports functionality for data secrecy in key distribution and data integrity through message authentication codes.

In a symmetric key system, the secret key in each smart card should be unique so that discovery of the key does not negotiate the whole system. This key diversification supports a unique cryptographic key for each smart card during personalization. The system develop the diversified key from the set of a system master key and a unique card characteristic including the microprocessor serial number.[45]

The CAD include a copy of the master key so they can manufacture the changed key for each smart card. Compromise of a diversified key has small effect on the system, supported the compromise is find and the card can be eliminated from the system. The compromise of a master key can have serious result.

The compromise of a master key needed that all cards should be loaded with new keys, which is costly to the systems and cumbersome for users. It can avoid key compromise, the CAD can include a security module with secured memory (i.e., smart card), where the master key is stored.

[**Biometric Authentication**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

## **What is Biometric Authentication?**

Authentication is a way to verify, beyond a doubt, that a person is who they say they are. Biometric authentication performs this verification by checking distinctive biological or behavioral characteristics.[5]

An authentication system works by comparing provided data with validated user information stored in a database. In traditional systems, this information is passwords. In biometric authentication, this information is defined as physical or behavioral traits.

For example, in a facial recognition system, different facial features are processed and converted into numerical data, which is stored in a database. When a person tries to log in, the system recaptures their face, extracts numerical data, and then compares it with what’s stored in the database. Other types of biometric authentication are:

• Fingerprint scanning

• DNA matching

• Retina scanning

• Vein scanning

• Behavioral biometrics [31]

Behavioral biometrics verify identity by analyzing physical and cognitive behavior of a user. They use machine learning algorithms to determine patterns in user behavior and activities. These patterns are then used to detect whether someone is who they say they are.

Examples of behavioral biometrics are:

• Touchscreen use (how much area of the screen are they using)

• Typing dynamics (keyboard shortcuts or typing speed)

• Mouse activity

[**Authorization**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

Authorization is the procedure of permitting someone to do something. It defines it an approach to check if the user has permission to need a resource or not. It can represent that what data and information one user can access.[11]

It is also called as AuthZ. The authorization generally works with authentication so that the system can understand who is accessing the information. Authorization is a security structure used to decide user/client privileges or access levels associated with system resources, such as computer programs, files, services, data and application features.

Authorization is generally preceded by authentication for customer identity verification. System administrators (SA) are generally assigned permission levels covering some system and customer resources.[40]

During authorization, a system checks an authenticated user's access rules and either grants or waste resource access. Modern and multiuser operating systems based on efficiently designed authorization processes to support application deployment and administration.

Key factors such as user type, number, and credentials needing verification and associated actions and roles. For instance, role-based authorization can be designated by user groups needing definite user resource tracking privileges.

Moreover, authorization can be based on an enterprise authentication structure, such as Active Directory (AD), for seamless security policy integration. For instance, ASP.NET works with Internet Information Server (IIS) and Microsoft Windows to support authentication and authorization services for internet-based .NET applications.[40]

Windows uses New Technology File System (NTFS) to support Access Control Lists (ACL) for some resources. The ACL serves as the final authority on resource access. The .NET Framework supports an alternate role-based security method for authorization support.

[**API Keys**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

“What is an API key?”. [Application programming interfaces (APIs)](https://blog.hubspot.com/website/application-programming-interface-api?_ga=2.175100706.1030540387.1662625458-1414316832.1662625458&hubs_content=blog.hubspot.com/website/api-keys&hubs_content-cta=Application%20programming%20interfaces%20(APIs)) allow software programs to interact, share data, and integrate their functionalities. APIs facilitate conversations between otherwise disconnected software and are the technology behind many powerful integrations you use daily. [30]

Unlike a face-to-face conversation, however, it’s harder for an API to verify whether the app it's talking to is who it claims to be. Since they often expose private and sensitive information, APIs need ways of identifying their clients before letting them through.

Otherwise, it’s a big security gamble.

API keys are one such security measure — they act like an ID card for the client making an API request, helping APIs assign the proper access permissions and track how their data are being used.

[**Basic Auth**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

## **What is Basic Authentication?**

Used widely in HTTP-based communication, basic method is the means of authenticating end-users before granting access to resources or communication. For the same, intended users are instructed to deliver primary credentials like user names and login passwords.

In this method, the base-64 encoded data is transmitted through an Authorization Header.

As it doesn’t demand cookies, login pages, and other identifiers to come into action, it’s considered as the most austere user-authentication technique helping in achieving full access control. [10]

[**HMAC**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

HMAC (Hash-based Message Authentication Code) is a type of a message authentication code (MAC) that is acquired by executing a cryptographic hash function on the data (that is) to be authenticated and a secret shared key. Like any of the MAC, it is used for both data integrity and authentication. Checking data integrity is necessary for the parties involved in communication. HTTPS, SFTP, FTPS, and other transfer protocols use HMAC. The cryptographic hash function may be MD-5, SHA-1, or SHA-256. Digital signatures are nearly similar to HMACs i.e they both employ a hash function and a shared key. The difference lies in the keys i.e HMACs use symmetric key(same copy) while Signatures use asymmetric (two different keys).

[**OAuth**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)

OAuth is an open-standard authorization protocol or framework that describes how unrelated servers and services can safely allow authenticated access to their assets without actually sharing the initial, related, single logon credential. In authentication parlance, this is known as secure, third-party, user-agent, delegated authorization.

**Conclusions**

Authentication and authorisation methods play important roles in securing various digital operations that people are directly affected such as people who use digital tools. They help digital tools to identify real identities of people (as users), they can be used both to encrypt and decrypt data and so on. These actions help platforms to identify the real persons who have official\real access to private data that can be only reached by the exact people. One of the way of authentication is biometric authentication which verifies people by checking distinctive biological or behavioral characteristics.

Authorization is the procedure of permitting someone to do something. It defines it as an approach to check if the user has permission to need a resource or not. It can represent that what data and information one user can access. During authorization, a system checks an authenticated user's access rules and either grants or waste resource access. Considering the modern and multiuser operating systems of today’s organisations, it is necessary to efficiently design authorization processes to support application deployment and administration.

**3**[**.4**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.7pii0khen2cd) **Cyber-Resilience Implementation**

**Redundancy**

Redundancy is an engineering term which means “the duplication of critical components or functions of a system with the intention of increasing reliability of the system, usually in the form of a backup or fail-safe, or to improve actual system performance”

In computer science there are four major forms of redundancy:

Hardware

Software

Information

Time

When we talk about building redundancy into an infrastructure project, we will most likely be referring to hardware redundancy. Hardware redundancy simply means adding a duplicate device or component within the system. This steps in when a primary device or component fails. The goal is to ensure zero downtime.[34]

**Server redundancy**

In some businesses the computing environment includes server redundancy. To enable this, a replica of the server is created with the same computing power, storage, applications and other operational parameters.

A redundant server is kept offline. That is, it powers on with network/Internet connectivity but is not used as a ‘live server’. In case of failure, downtime or excessive traffic at the primary server, a redundant server can be implemented to take the primary server’s place or share its load. [33]

**Network redundancy**

Network redundancy is a process through which additional or alternate instances of network devices, equipment or communication mediums are installed within network infrastructure. It is a method for ensuring network availability in case of a network device or path failure and unavailability. As such, it provides a means of network failover.

Network redundancy is primarily implemented in enterprise network infrastructure to provide a redundant source of network communications. It serves as a backup mechanism for quickly swapping network operations onto redundant infrastructure in the event of unplanned network outages. [33]

**Internet Redundancy**

One form of network redundancy is Multi-Wan, which is available through some firewalls such as the [WatchGuard UTM Firewalls](https://www.your-itdepartment.co.uk/it-security/firewall-as-a-service/) that we at Your IT provide. Multi-Wan provides the ability to have multiple internet connections into a single site. If one internet connection goes down you simply switch to another giving a continuous connection.

**Backup Types**

**3 Main Types of Data Backup Strategies**

Let’s review the advantages and disadvantages of the main types of backup methods - full, differential, and incremental - to ensure you choose the optimal data backup strategy for your organization’s needs.

**1. Full Data Backups**

As the name implies, a full data backup refers to a complete copy of all data, including files and folders, regardless of any previous backups or circumstances. The entire data set is backed up and stored either onsite, at another location, or both. Full backups are the quickest to restore since all files are contained in the same backup set.[29]

Full backups typically provide the best protection against critical data loss, but they are often conducted periodically on a predetermined schedule because of the time and expense involved. However, any good backup plan should include a full data backup, at least once, as a foundation.

A full data backup is generally the easiest to perform but not necessarily the simplest to administer. Here are a few disadvantages to keep in mind:

Uses the most storage out of each method

Requires substantial network bandwidth

Often needs additional hardware if the data set is large

Considerable implementation time, depending on the size of the business

The other two backup types — differential and incremental — are offshoots of a full data backup in that they work with a full data backup. The differential and incremental backups are similar and have some subtle yet important differences.[18]

**2. Differential Data Backup**

Differential backups start with a full backup to store all files. Then, differential backups are run to include all the changes made to files and folders since the last full backup. A differential backup is cumulative. Therefore, a full and a differential backup together include all the files in your data set, changed and unchanged.

Companies can restore data faster than a full backup since differential backups require only two components: the initial full backup and the most recent differential backup. While differential backups enable a faster restoration time over incremental backups, they need more space and use much more network bandwidth. [39]

**3. Incremental Data Backup**

Similar to differential backups, incremental backups start with a full backup to store all files. Then, the subsequent backups only store additions and alterations made to the previous backup or the most recent incremental backup. While a differential backup contains all data since the last full backup, an incremental backup includes only the data which has changed since any previous backup.

Since incremental backups only store the changes (increments), they require the least storage space, enabling accelerated backup speeds. However, they have the most time-consuming restoration process. To restore a full system, a company first needs to restore the latest full backup and then restore each incremental backup in order. Complete restoration becomes increasingly difficult or impossible if one incremental backup is lost, damaged, or stolen.[47]

**Conclusions**

Cyber resilience implementation strategies are important subjects to protect organisations from various cyber-attacks, such as loss of data, reputation, resources, money and trust, where cyber resilience implementations help such organisations to stay safe as much as possible via server redundancy, network redundancy and internet redundancy and data backup strategies.

**3.5 Securing Embedded Systems**

**Embedded Systems**

Embedded systems security is a strategy that provides embedded systems protection from

cyber attacks. [Embedded systems](https://www.techslang.com/definition/what-is-an-embedded-system/), also known as “embedded computers,” are small devices

with a dedicated function within a more extensive system.

[Internet of Things (IoT) devices](https://www.techslang.com/definition/what-is-the-internet-of-things-iot/) are examples of embedded systems. For instance, [smart](https://www.techslang.com/intelligent-home-are-we-ready-to-move-in/)

[household appliances](https://www.techslang.com/intelligent-home-are-we-ready-to-move-in/), such as your refrigerator, thermostat, and security alarm perform

different functions. However, they are part of the whole household system and

interconnected to each other through the Internet.Embedded systems security aims to protect the software running on these devices since malicious actors are increasingly targeting them.

**How Does Embedded Systems Security Work?**

While embedded systems security is a strategy, it comprises different security systems and

tools. There are several providers of embedded systems security. Kaspersky Embedded

Systems Security, for example, protects Windows embedded systems. Star Lab is another

provider specializing in protecting embedded systems in the aerospace and military sectors.

Regardless of the company or provider, embedded systems security has crucial aspects,

which include:

* Threat monitoring: Embedded systems security should continuously track all parts of the embedded system for potential threats. A patch should immediately be designed and deployed when a threat is detected.
* Continuous security design improvement: Embedded systems security works on the security design endlessly as new attack methods emerge. Even when the cyber attack tactic is not yet detected on the embedded system, the security design must be updated to protect against new malicious tactics.
* Regular security upgrade: It’s critical to frequently upgrade embedded systems software to protect devices from emerging threats.
* 24×7 support: Providers of embedded systems security should offer round-the-clock support to help internal teams with integration, testing, and other processes involved in the embedded system operations.

**IoT**

IoT and ICS devices are considered end points. In other words, they are devices at the end of a communications chain that starts with a person or robotics device, and ends in cloud platforms and data centers. IoT and ICS devices don’t just appear out of thin air. They are designed, developed and managed, just like any other computer. Figure 1 provides an overview of the elements inside of a typical IoT device.[43]

Any IoT device has each of these elements inside:

* Firmware: Read-only memory embedded in the device that provides low-level control of the hardware. It can be updated, but usually not programmed. Communicates between each of the elements in the device as well as other networked devices.
* Protective services: A portion of the device’s firmware or operating system that provides security functionality, including the ability to isolate processes so that they can’t be used to defeat security, and encryption.
* Motion sensors: The combined hardware and software used to track how the device is moved. Can include detection of simple movement (e.g., moving the device back and forth, or up and down), or satellite connectivity (e.g., Global Positioning System or GLONASS).
* Microcontroller: The processor used to run the software and provide the “brains” of the unit.
* Connectivity stack: Responsible for providing network connectivity. Networking capabilities can include Bluetooth, mobile (e.g., 3G, 4G, 5G), Zigbee, LoRA, SigFox or WiFi.
* Authentication services: When included, provides the ability for the IoT device to verify and validate users, network traffic or processes.
* Power management: If the device requires significant use of power, functionality that manages power usage, as well as the charging of the device.
* Battery/power: The physical capability to store power, as well as receive power from a remote source.
* Memory: The working “muscle” of the IoT device, in that it provides the ability to store working data, machine code and information that is then addressed by the processor.
* Storage: Provides the ability to capture and keep data for relatively long periods of time. Such information can include storing the location(s) the IoT device wearer/operator took the device, information about other devices that have connected with it and information entered by the user. Such information can be added actively (e.g., by the user programming the IoT/OT device on purpose), or passively (e.g., the device capturing the motions and actions of the wearer/user).[43, 46]

**Conclusions**

Embedded systems security is a strategy that provides embedded systems protection from

cyber attacks. [Embedded systems](https://www.techslang.com/definition/what-is-an-embedded-system/), also known as “embedded computers,” are small devices

with a dedicated function within a more extensive system that provide 24\7 security services.

In addition, IoT and ICS devices are considered end points. In other words, they are devices at the end of a communications chain that starts with a person or robotics device, and ends in cloud platforms and data centers. Both IoT and ICS devices are designed, developed and managed, just like any other computers with various components including firmware, microcontroller, motion sensors, battery/power, memory, storage etc.

**3.6 Basics Cryptographics Concepts**

**Digital Signatures**

A digital signature is a mathematical technique which validates the authenticity and integrity of a message, software or digital documents. It allows us to verify the author name, date and time of signatures, and authenticate the message contents. The digital signature offers far more inherent security and intended to solve the problem of tampering and impersonation (Intentionally copy another person's characteristics) in digital communications.[1]

The computer-based business information authentication interrelates both technology and the law. It also calls for cooperation between the people of different professional backgrounds and areas of expertise. The digital signatures are different from other electronic signatures not only in terms of process and result, but also it makes digital signatures more serviceable for legal purposes. Some electronic signatures that legally recognizable as signatures may not be secure as digital signatures and may lead to uncertainty and disputes.[14]

**Application of Digital Signature**

The important reason to implement digital signature to communication is:

Authentication

Non-repudiation

Integrity

**Authentication**

Authentication is a process which verifies the identity of a user who wants to access the system. In the digital signature, authentication helps to authenticate the sources of messages.[7]

**Non-repudiation**

Non-repudiation means assurance of something that cannot be denied. It ensures that someone to a contract or communication cannot later deny the authenticity of their signature on a document or in a file or the sending of a message that they originated.

**Integrity**

Integrity ensures that the message is real, accurate and safeguards from unauthorized user modification during the transmission.

**Key Exchange**

The Internet Key Exchange is a hybrid protocol that is often used for key management purposes in IPSec networks. It is often used as a method of exchanging encryption keys and/or authentication keys through an unsecured medium like the Internet. In other words, the Internet Key Exchange aims to provide safe and secure encryption for unsecure or vulnerable environments.[19, 40]

**What are the benefits of the Internet Key Exchange?**

The Internet Key Exchange offers numerous additional benefits including flexibility. Below you can find some of these benefits:

Internet Key Exchange offers the change to change encryption during IPsec sessions.

Through the use of Internet Key Exchange, the need for manual specification of all the IPSec security parameters is eliminated.

Internet Key Exchange allows certification authority, as a result it offers an additional layer of security.

A specific lifetime can be set for IPsec security association when the Internet Key Exchange is used.

Internet Key Exchange permits the dynamic authentication of peers.[52]

**Symmetric and Asymmetric**

**Symmetric**

This is the simplest kind of encryption that involves only one secret key to cipher and decipher information. Symmetric encryption is an old and best-known technique. It uses a secret key that can either be a number, a word or a string of random letters. It is a blended with the plain text of a message to change the content in a particular way. The sender and the recipient should know the secret key that is used to encrypt and decrypt all the messages. Blowfish, AES, RC4, DES, RC5, and RC6 are examples of symmetric encryption. The most widely used symmetric algorithm is AES-128, AES-192, and AES-256.

The main disadvantage of the symmetric key encryption is that all parties involved have to exchange the key used to encrypt the data before they can decrypt it. [20, 37]

**Pros and Cons of Symmetric Encryption**

**Pros**

Faster: Since it’s using a single key for encryption and decryption, it’s faster to execute.

Identity verification: It uses password authentication as a security purpose to prove the receiver’s identity.

Easy to execute & manage: Users have only one key for encryption and decryption so it’s easy to execute and manage.[44]

**Cons**

The chances of sharing encryption keys securely are less; it is difficult and challenging to share keys in Symmetric Encryption.

Symmetric is not that scalable, as it’s not suitable for various users.

**Asymmetric**

Asymmetric encryption is also known as public key cryptography, which is a relatively new method, compared to symmetric encryption. Asymmetric encryption uses two keys to encrypt a plain text. Secret keys are exchanged over the Internet or a large network. It ensures that malicious persons do not misuse the keys. It is important to note that anyone with a secret key can decrypt the message and this is why asymmetric encryption uses two related keys to boosting security. A [public key](https://www.ssl2buy.com/wiki/what-is-a-public-and-private-key-pair/) is made freely available to anyone who might want to send you a message. The second private key is kept a secret so that you can only know. [41]

A message that is encrypted using a public key can only be decrypted using a private key, while also, a message encrypted using a private key can be decrypted using a public key. Security of the public key is not required because it is publicly available and can be passed over the internet. Asymmetric key has a far better power in ensuring the security of information transmitted during communication.[41, 51]

Asymmetric encryption is mostly used in day-to-day communication channels, especially over the Internet. Popular asymmetric key encryption algorithm includes EIGamal, [RSA, DSA, Elliptic curve techniques](https://www.ssl2buy.com/wiki/ecc-algorithm-to-enhance-security-with-better-key-strength/), PKCS.

**Pros and Cons of Asymmetric Encryption**

**Pros**

Asymmetric Encryption has two keys, one public and one private, so there’s no problem with distributing keys.

Again, with a pair of keys, it is not difficult to communicate with multiple parties and that’s how it is more scalable in large networks.

**Cons**

Performance: Asymmetric Encryption is slower in performance compared to Symmetric Encryption.

Hard to understand and execute: Asymmetric Encryption is not that easy to implement and manage due to its large key sizes. [17]

**Asymmetric Encryption in Digital Certificates**

To use asymmetric encryption, there must be a way of discovering public keys. One typical technique is using digital certificates in a client-server model of communication. A certificate is a package of information that identifies a user and a server. It contains information such as an organization’s name, the organization that issued the certificate, the users’ email address and country, and users public key.

When a server and a client require a secure encrypted communication, they send a query over the network to the other party, which sends back a copy of the certificate. The other party’s public key can be extracted from the certificate. A certificate can also be used to uniquely identify the holder.

SSL/TLS uses both asymmetric and symmetric encryption, quickly look at digitally signed [SSL certificates](https://www.ssl2buy.com/) issued by trusted certificate authorities (CAs).

**Difference Between Symmetric and Asymmetric Encryption**

Symmetric encryption uses a single key that needs to be shared among the people who need to receive the message while asymmetric encryption uses a pair of public key and a private key to encrypt and decrypt messages when communicating.

Symmetric encryption is an old technique while asymmetric encryption is relatively new.

Asymmetric encryption was introduced to complement the inherent problem of the need to share the key in symmetric encryption model, eliminating the need to share the key by using a pair of public-private keys.

Asymmetric encryption takes relatively more time than the symmetric encryption. [26]

**Transport encryption**

Transport Layer Security (TLS) is a type of encryption protocol used to provide communication security over a network.

Websites use TLS to secure all communications between their servers and web browsers.

Encryption is an enabler to achieve compliance and data privacy--the utmost requirements of business environment. It helps organizations retain control over data, protect sensitive information and ensure the security of communications. With encryption, even if unauthorized users manage to access sensitive files, they won't be able to read the data.[26]

**What is TLS?**

Transport Layer Security (TLS) is a type of encryption protocol used to provide communication security over a network. Websites use TLS to secure all communications between their servers and web browsers. Careful configuration of TLS also provides additional privacy properties--like ensuring future disclosure of encryption keys cannot be used to decrypt any TLS communications recorded in the past.[26, 37]

**How does it work?**

When the connection between a client and a server is secured by TLS, it has the following properties:

1) It uses the same cryptographic keys for encryption and decryption of information.

2) The identity of the communicating parties can be authenticated using public-key cryptography.

3) Messages transmitted through TLS security will include integrity check using a message authentication code to prevent undetected loss or alteration of the data during transmission.

**Hashing**

**What is Hashing in Cybersecurity?**

In computer science and cryptography, a hash function is a deterministic procedure that takes an input (or “message”) and returns a string of characters of a fixed size—which is usually a “digest”—that is unique to the input.

A hash function is used in many cybersecurity algorithms and protocols, such as password storage and digital signature. Hashing is also used in a data structure, such as a hash table (a data structure that stores data), for a quick search and insertion. [36]

**The Purpose of Hashing**

Learning the answer to the question about what is hashing in cybersecurity can help a professional use hashing algorithms for data encryption and data security. Cybersecurity professionals convert a large block of input data using the hashing algorithm into a smaller fixed-length string as the final output.

Businesses always want to secure their data servers and cloud storage systems from vulnerabilities to malicious software. Hashing helps cybersecurity professionals ensure that the data stored on servers and cloud storage systems remains unreadable by hackers.[23]

**What is Hashing Used for?**

Hashing is a one-way function that turns a file or string of text into a unique digest of the message. The hash value is calculated by a hashing algorithm using the binary data of a particular file. Now let’s look at the different uses that hashing has in cybersecurity.

**Storage Password**

Hashes provide security to an organization’s cyber system so that hackers cannot steal it; for example, email passwords stored on servers.

**Digital Signatures**

Hashing is a way to encrypt and decrypt digital signatures, verifying the message’s sender and receiver.

**Document Management**

The authenticity of data can be verified with the use of hashing algorithms. When a document is entirely written, the cybersecurity specialist will use a hash to secure it.

**File Management**

Businesses use hashes to index data, recognize files, and erase duplicate files. An organization can save significant time utilizing hashes when working with a cyber system with thousands of files.

**Steganography**

A steganography technique involves hiding sensitive information within an ordinary, non-secret file or message, so that it will not be detected. The sensitive information will then be extracted from the ordinary file or message at its destination, thus avoiding detection.

Steganography is an additional step that can be used in conjunction with encryption in order to conceal or protect data.

Steganography is a means of concealing secret information within (or even on top of) an otherwise mundane, non-secret document or other media to avoid detection. It comes from the Greek words steganos, which means “covered” or “hidden,” and graph, which means “to write.” Hence, “hidden writing.”

You can use steganography to hide text, video, images, or even audio data. It’s a helpful bit of knowledge, limited only by the type of medium and the author’s imagination.[9]

**Conclusions**

Cryptographic concepts became an important area of cybersecurity where they validate the authenticity and integrity of a message, software or digital documents. It allows us to verify the author name, date and time of signatures, and authenticate the message contents such as the digital signatures, which offers far more inherent security and intended to solve the problem of tampering and impersonation (Intentionally copy another person's characteristics) in digital communications.

Similarly, Transport Layer Security (TLS) is a type of encryption protocol used to provide communication security over a network. Websites use TLS to secure all communications between their servers and web browsers. Careful configuration of TLS also provides additional privacy properties like ensuring future disclosure of encryption keys cannot be used to decrypt any TLS communications recorded in the past.

[**The Geek’s Corner**](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.floxyjcfjlz3)

**XML External Entities (XEE)**

For those who are Geek, XML can be a subject of study where this is an attack against a web application that parses XML\* input. This input can reference an external entity, attempting to exploit a vulnerability in the parser. An ‘external entity’ in this context refers to a storage unit, such as a hard drive. An XML parser can be duped into sending data to an unauthorized external entity, which can pass sensitive data directly to an attacker. [41]

The best ways to prevent XEE attacks are to have web applications accept a less complex type of data, such as JSON\*\*, or at the very least to patch XML parsers and disable the use of external entities in an XML application.

\*XML or Extensible Markup Language is a markup language intended to be both human-readable and machine-readable. Due to its complexity and security vulnerabilities, it is now being phased out of use in many web applications.

\*\*JavaScript Object Notation (JSON) is a type of simple, human-readable notation often used to transmit data over the internet. Although it was originally created for JavaScript, JSON is language-agnostic and can be interpreted by many different programming languages.

**For more;**

<https://www.imperva.com/learn/application-security/xxe-xml-external-entity/>

<https://brightsec.com/blog/xxe-vulnerability/>

<https://www.invicti.com/learn/xml-external-entity-xxe/>

[https://www.hackerone.com/knowledge-center/xxe-complete-guide-impact-examples-and-pr](https://www.hackerone.com/knowledge-center/xxe-complete-guide-impact-examples-and-prevention)

[evention](https://www.hackerone.com/knowledge-center/xxe-complete-guide-impact-examples-and-prevention)

**Conclusions**

In this unit, you have learned various ways to understand how to secure and design an IT system to meet its security requirements, balancing this with its functional requirements, how security architecture works as a unified security design that addresses the necessities and potential risks involved in a certain scenario or environment, when and where to apply security controls and more.

**Quiz**

| **1)** | **Cloud Security refers to** | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | |  |  |  |  |
|  | 1. A broad set of policies |  |  |  |  |  |
|  | 1. Technologies |  |  |  |  |  |
|  | 1. Deployment Controls |  |  |  |  |  |
|  | 1. **All of the above**   **2. Cloud security protects?**   1. Data 2. Infrastructure 3. Applications 4. **All of the above** |  |  |  |  |  |

| **3. Cloud providers ensure that ………………….via the cloud are secure by implementing testing and acceptance procedures for outsourced or packaged application code.** | | | | | |
| --- | --- | --- | --- | --- | --- |
|  |  | |  |  |  |
|  | 1. Platform as a service |  |  |  |  |
|  | 1. Infrastructure as a service |  |  |  |  |
|  | 1. **Applications available as a service** |  |  |  |  |
|  | 1. None of these   **4. The major security concerns of Cloud Computing are:?**   1. IT Leaders struggle to trust relatively new and unfamiliar cloud providers as part of their extended enterprises. 2. IT professionals question whether cloud providers have the levels of infrastructure security to be able to ward off cyber-attacks. 3. Providers have the mechanisms in place to be able to manage, measure, and report on industry regulations? And can they be accountable if they fail to comply? 4. IT managers are right to look for service level guarantees. But in the case of the cloud, response times cannot be guaranteed since data travels through the Internet 5. **All of the above** |  |  |  |  |

**References**

1. Abad-Segura, E.; González-Zamar, M.D.; Belmonte-Ureña, L.J. Effects of circular economy policies on the environment and sustainable growth: Worldwide research. Sustainability 2020, 12, 5792. [CrossRef]

2. Agbo, F.J.; Sanusi, I.T.; Oyelere, S.S.; Suhonen, J. Application of Virtual Reality in Computer Science Education: A Systemic Review Based on Bibliometric and Content Analysis Methods. Educ. Sci. 2021, 11, 142. [CrossRef]

3. Agbo, F.J.; Oyelere, S.S.; Suhonen, J.; Tukiainen, M. Scientific production and thematic breakthroughs in smart learning environments: A bibliometric analysis. Smart Learn. Environ. 2021, 8, 1–25. [CrossRef]

4. Ali, A.; Alourani, A. An Investigation of Cloud Computing and E-Learning for Educational Advancement. Int. J. Comput. Sci. Netw. Secur. 2021, 21, 216–222.

5. Alshaher, A.; Mustafa, M.; Abdulmuhsin, A.A. A multi-dimensional approach for accepting and using e-learning cloud: A developing country’s perspective. Int. J. Knowl. Manag. Stud. 2022, 13, 172–212. [CrossRef]

6. Andriamamonjy, A.; Saelens, D.; Klein, R. A combined scientometric and conventional literature review to grasp the entire BIM knowledge and its integration with energy simulation. J. Build. Eng. 2019, 22, 513–527. [CrossRef]

7. Ariza, J.; Jimeno, M.; Villanueva-Polanco, R.; Capacho, J. Provisioning Computational Resources for Cloud-Based e-Learning Platforms Using Deep Learning Techniques. IEEE Access 2021, 9, 89798–89811. [CrossRef]

8. Birkle, C.; Pendlebury, D.A.; Schnell, J.; Adams, J. WebofScienceasadatasourceforresearchonscientificandscholarlyactivity. Quant. Sci. Stud. 2020, 1, 363–376. [CrossRef]

9. Chen, X.; Zou, D.; Xie, H.; Wang, F.L. Past, present, and future of smart learning: A topic-based bibliometric analysis. Int. J. Educ. Technol. High. Educ. 2021, 18, 2. [CrossRef]

10. Daradkeh, M.; Abualigah, L.; Atalla, S.; Mansoor, W. Scientometric Analysis and Classification of Research Using Convolutional Neural Networks: A Case Study in Data Science and Analytics. Electronics 2022, 11, 2066. [CrossRef]

11. Degambur, L.N.; Armoogum, S.; Pudaruth, S. A Study of Security Impacts and Cryptographic Techniques in Cloud-based e-Learning Technologies. Int. J. Adv. Comput. Sci. Appl. 2022, 13, 58–66. [CrossRef]

12. Dima, A.; Bugheanu, A.M.; Dinulescu, R.; Potcovaru, A.M.; Stefanescu, C.A.; Marin, I. Exploring the Research Regarding Frugal Innovation and Business Sustainability through Bibliometric Analysis. Sustainability 2022, 14, 1326. [CrossRef]

13. Dima, A. The Importance of Innovation in Entrepreneurship for Economic Growth and Development. A Bibliometric Analysis. Rev. Int. Comp. Manag. 2021, 22, 120–131. [CrossRef]

14. Donthu, N.; Kumar, S.; Mukherjee, D.; Pandey, N.; Lim, W.M. How to conduct a bibliometric analysis: An overview and guidelines. J. Bus. Res. 2021, 133, 285–296. [CrossRef]

15. Donthu, N.; Kumar, S.; Pandey, N.; Gupta, P. Forty years of the International Journal of Information Management: A bibliometric analysis. Int..J Inform. Manag. 2021, 57, 102307. [CrossRef]

16. Forés, B.; Breithaupt Janssen, Z.; Takashi Kato, H. A Bibliometric Overview of Tourism Family Business. Sustainability 2021, 13, 12822. [CrossRef]

17. Garg, D.; Sidhu, J.; Rani, S. Emerging trends in cloud computing security: A bibliometric analyses. IET Softw. 2019, 13, 223–231. [CrossRef]

18. Gao J. Research on the Corporate Financial Transformation with Big Data Technologies. International Journal of Progressive Sciences and Technologies.2022; 32 (2): 08 - 12. http: //dx. doi. org/10.52155/ijpsat. v32.2.4316

19. Gao, Jun. (2022). Research on the Corporate Financial Transformation with Big Data Technologies.32.8 - 12.10.52155/ijpsat. v32.2.4316.

20. Gao, J. (2022). Analysis of the Financial Internal Control Strategies of SME Based on the Background of Big Data. Technium Social Sciences Journal, 32 (1), 352–358. https: //doi. org/10.47577/tssj. v32i1.6502

21. Gao, Jun. (2022). Research on Earnings Management under IFRS Framework. International Journal of Science and Research (IJSR).11.1617 - 1622.10.21275/SR22625101243.

22. Gao, Jun. (2022). Research on Financial Informatization Construction of Business and Finance Integration. International Journal of Science and Research (IJSR).11.1272 - 1276.10.21275/SR22704192347.

23. Gupta H and Sharma S. Security Challenges in Adopting Internet of Things for Smart Network. In: 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 18–19 June 2021, pp. 761–765.

24.Handa A, Sharma A and Shukla S.K. Machine learning in cybersecurity: a review. Wiley Interdiscip Rev Data Min Knowl Discov 2019; 9: e1306.

25. Hensel, P.G. Reproducibility and replicability crisis: How management compares to psychology and economics—A systematic review of literature. Eur. Manag. J. 2021, 39, 577–594. [CrossRef]

26. Huang M, Liu A, Xiong NN, et al. An effective serviceoriented networking management architecture for 5G-enabled internet of things. Comput Networks 2020; 173: 107208.

27. Jangjou M and Sohrabi MK. A comprehensive survey on security challenges in different network layers in cloud computing. Arch Comput Methods Eng 2022: 1–22.

28. Khedr, A.E.; Idrees, A.M.; Salem, R. Enhancing the e-learning system based on a novel tasks’ classification load-balancing algorithm. PeerJ Comput. Sci. 2021, 7, e669. [CrossRef] [PubMed]

29. Khraisat A, Gondal P, Vamplew P, et al. Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine. Electronics 2020;9: 173.

30. Kumar Bhardwaj, A.; Garg, L.; Garg, A.; Gajpal, Y. E-learning during covid-19 outbreak: Cloud computing adoption in indian public universities. Comput. Mater. Contin. 2021, 66, 2471–2492. [CrossRef]

31. Lei, T.T.; Qin, Y.; Xu, Z.S.; Borzooei, R.A. A bibliometric analysis of Iranian Journal of Fuzzy Systems (2007–2020). Iran J. Fuzzy Syst. 2021, 18, 1–17. [CrossRef]

32. Liu SHUANG. Research on strengthening financial control of real estate enterprises under the new situation [J]. China Collective Economy, 2021 (33): 146 - 147.

33. Liu Z, Sampaio P, Pishchulov G, et al. The architectural design and implementation of a digital platform for Industry 4.0 SME collaboration. Comput Industry 2022; 138: 103623.

34. Mahdavisharif M, Jamali S and Fotohi R. Big data-aware intrusion detection system in communication networks: a deep learning approach. J Grid Comput 2021; 19(4): 1–28.

35. Mostafa, N.; Hamdy, W.; Alawady, H. Impacts of Internet of Things on Supply Chains: A Framework for Warehousing. Soc. Sci. 2019, 8, 84. [CrossRef]

36. Nguyen, D.T.; Vu, T.H.N.; Kim, H.T. Factors affecting e-learning based cloud computing acceptance: An empirical study at Vietnamese universities. Int. J. Econ. Manag. 2021, 20, 118–133. [CrossRef]

37. Paul, J.; Lim, W.M.; O’Cass, A.; Hao, A.W.; Bresciani, S. Scientific procedures and rationales for systematic literature reviews (SPAR-4-SLR). Int. J. Consum. Stud. 2021, 45, O1–O16. [CrossRef]

38. Pranckute, R. Web of Science (WoS) and Scopus: The Titans of Bibliographic Information in Today’s Academic World. ˙ Publications 2021, 9, 12. [CrossRef]

39. Prabhavathy, M.; Uma Maheswari, S. Prevention of runtime malware injection attack in cloud using unsupervised learning. Intell. Autom. Soft Comput. 2022, 32, 101–114. [CrossRef]