**CYBERYOUTH**

**Formazione non formale per la sicurezza informatica e la resilienza delle organizzazioni giovanili e dei giovani**

***Accademia online per la cybersecurity giovanile***

****

**Design & Architettura**

**dei Sistemi di Sicurezza**

**Indice**

[Introduzione](#_heading=h.fpb3yt6tcffi)

[Cosa imparerai](#_heading=h.bd1t10x6k115)

[Perché è importante e come può aiutare le organizzazioni giovanili e i giovani nella vita di tutti i giorni](#_heading=h.1fqj6c1n3ygs)

[Che carriera puoi intraprendere](#_heading=h.l25siuf1d90q)

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

[Materiale](#_heading=h.mqxaktyycq90)

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

[Cloud Computing](#_heading=h.9s83abu6k8mw)

[Cloud Providers](#_heading=h.tgj5q1yodxt3)

[La virtualizzazione nel cloud computing e i tipi](#_heading=h.8dqpyeaepn7l)

[La virtualizzazione](#_heading=h.2edutmnfnzih)

[3.2 Secure Application Development &](#_heading=h.7y2eb4eisgml)

[Deployment](#_heading=h.pj4074oiiznu)

[Secure DevOps](#_heading=h.e8c04d1vks6m)

[Provisioning e Deprovisioning](#_heading=h.3e8j3fv7tix2)

[Tecniche di Secure Coding](#_heading=h.vj0nax6e07pk)

[Code Quality & Security](#_heading=h.2iin38dfua68)

[OSWAP](#_heading=h.kdqs1c1cebjy)

[Conclusioni](#_heading=h.m9w3qz8kqd83)

[3.3 Autemticazione & Autorizzazione](#_heading=h.kd7a4v2e48ko)

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

[Autenticazione Biometrica](#_heading=h.mzumzkjc3dq1)

[Biometria comportamentale](#_heading=h.c3zu471ir8ud)

[Autorizzazione](#_heading=h.rljua8oq530x)

[API Keys](#_heading=h.j0k89j22ijf0)

[Basic Auth](#_heading=h.522p06ufpnfw)

[HMAC](#_heading=h.6nk83loofql2)

[OAuth](#_heading=h.wnboata43vgi)

[3.4 Cyber-Resilience Implementation](#_heading=h.3ldkj8672fsb)

[Ridondanza](#_heading=h.ggkez0viczh5)

[Tipi di Backup](#_heading=h.e3l3oxpv3d3b)

[Conclusioni](#_heading=h.1uv5l5bha9dl)

[3.5 Securing Embedded Systems](#_heading=h.6vb8k98aoafu)

[Come funziona la sicurezza dei sistemi embedded?](#_heading=h.xd6e2a94xi3f)

[Conclusioni](#_heading=h.i25aewf57n1x)

[3.6 Concetti Base di Crittografia](#_heading=h.gl4mrx6xncs0)

[Firme Digitali](#_heading=h.s44114j6qhhw)

[Hashing](#_heading=h.fpl9na836hc7)

[Conclusioni](#_heading=h.4zaz98lhnyqm)

[L'angolo del geek](#_heading=h.nq1nuvhy0lye)

[Conclusioni](#_heading=h.na4y5vskbuk)

[Quiz](#_heading=h.imoqqav4b822)

[Fonti](#_heading=h.cuwifdumly93)

# 

# 

# 

# 

# **Introduzione**

## **Cosa imparerai**

Nell'unità, imparerai diverse modalità per capire come proteggere e progettare un sistema IT per soddisfare i suoi requisiti di sicurezza, bilanciando ciò con i suoi requisiti funzionali. Apprenderai come funziona l'architettura della sicurezza come un progetto unitario di sicurezza che affronta le necessità e i rischi potenziali coinvolti in uno scenario o ambiente specifico, quando e dove applicare i controlli di sicurezza e altro ancora.

## **Perché è importante e come può aiutare le organizzazioni giovanili e i giovani nella vita di tutti i giorni**

La progettazione e l'architettura sicure del sistema sono importanti perché risolvono problemi complessi di sicurezza selezionando le migliori soluzioni disponibili da una serie di componenti e strutture tecnologiche. Le decisioni prese determinano fondamentalmente se un'organizzazione può gestire in modo sicuro i propri dati, sistemi informativi e reti di comunicazione. L'architettura della sicurezza può tradurre ciascun requisito unico in strategie eseguibili e sviluppare un ambiente privo di rischi per un'organizzazione o un'azienda, allineandosi con gli ultimi standard di sicurezza e le esigenze organizzative/aziendali.

Un'altra importanza è il principio della separazione dei privilegi, spesso implementato suddividendo un sistema in diversi livelli, ognuno con il proprio insieme di privilegi. Questo principio rappresenta una parte importante della progettazione della sicurezza quando si progetta qualsiasi sistema ed è un approccio fondamentale.

## **Che carriera puoi intraprendere**

Tra i molteplici titoli di carriera, alcune delle carriere più importanti sono:

Ingegnere della sicurezza, che costruisce sistemi di sicurezza implementando e monitorando controlli di sicurezza con l'obiettivo di proteggere i dati di un'organizzazione da attacchi informatici, perdite o accessi non autorizzati.

Per esplicitare ulteriormente, ecco alcune delle mansioni e delle responsabilità trovate nelle offerte di lavoro di veri ingegneri della sicurezza su LinkedIn di Coursera.org:

- Identificare misure di sicurezza per migliorare la risposta agli incidenti

- Rispondere agli incidenti di sicurezza

- Eseguire valutazioni di sicurezza e audit del codice

- Sviluppare soluzioni tecniche per vulnerabilità di sicurezza

- Ricerca di nuovi vettori di attacco e sviluppo di modelli di minaccia

- Automatizzare miglioramenti in materia di sicurezza

Inoltre, ci sono altre carriere come ingegnere della sicurezza delle applicazioni, ingegnere cloud o architetto cloud, che sono tutte correlate al principale settore della sicurezza informatica.

## **Pre-requisiti**

Per intraprendere una carriera nel campo della sicurezza, ecco alcune aree di base (pre-requisite) e gradi che qualcuno dovrebbe possedere:

Lauree: Laurea in Informatica, IT, Ingegneria dei Sistemi o sicurezza informatica, o in un campo simile.

Gli ingegneri della sicurezza possono anche lavorare in ambienti che richiedono autorizzazioni di sicurezza rilasciate dal governo o certificazioni industriali come:

- Certified Information Systems Security Professional (CISSP)

- Certificazione SANS/GIAC (varie)

- Certified Information Security Manager (CISM)

- CompTIA Security+

- Certified Information Systems Auditor (CISA)

Secondo uno studio sugli attuali ingegneri della sicurezza informatica, ci sono alcune competenze e esperienze fondamentali in comune, tra cui:

- Una comprensione del codice informatico, in particolare di come appare il codice dannoso come virus o malware e come affrontarlo.

- Conoscenza delle tecnologie e dei metodi di valutazione dei rischi.

- Comprensione della computer forensics e dei protocolli di violazione della sicurezza.

- La capacità di effettuare valutazioni e valutazioni dei rischi di sicurezza.

- Esperienza nello sviluppo e nell'attuazione di procedure e politiche di sicurezza.

- Comprensione del software antivirus, della manutenzione dei firewall e della rilevazione degli hacker.

Altre competenze richieste per gli ingegneri della sicurezza informatica, come dichiarato da Cyberseek:

- Sicurezza delle informazioni

- Sicurezza delle reti

- Linux

- Sistemi informativi

- Python

- Crittografia

- Cisco

- Autenticazione

# **Materiale**

# 

# **3.1 Cloud Models**

## Cloud Computing

In termini molto semplici, il cloud computing significa archiviare e accedere ai dati e ai programmi su server remoti ospitati su Internet, anziché sull'hard disk del computer o su un server locale. Il cloud computing è anche noto come informatica basata su Internet.

**Architettura del Cloud Computing:** L'architettura del cloud computing si riferisce ai componenti e ai sotto-componenti necessari per il cloud computing. Questi componenti si riferiscono tipicamente a:

1. Front end (cliente pesante, cliente leggero)

2. Piattaforme backend (server, archiviazione)

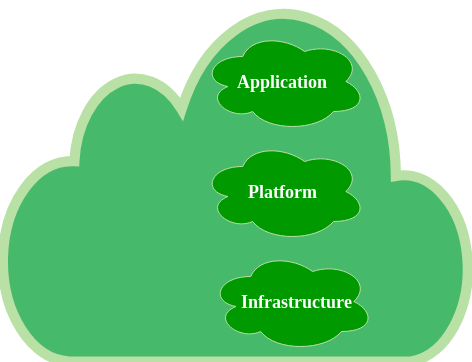
3. Consegna basata su cloud e una rete (Internet, Intranet, Intercloud).

**Ospitare un cloud:** Nel cloud computing, ci sono tre livelli. Le aziende utilizzano questi livelli in base al servizio che forniscono.

1. Infrastruttura

2. Piattaforma

3. Applicazione



I TRE LIVELLI DEL CLOUD COMPUTING [4]

**Cos’è l’Hosting**

In termini molto semplici, l'hosting è il processo di esternalizzare le risorse di archiviazione di un'organizzazione a un fornitore di servizi che offre i suoi servizi di infrastruttura in un modello di utilità.

**Vantaggi dell'Hosting basato su Cloud:**

1. **Scalabilità:** Con l'hosting basato su cloud, è facile aumentare o ridurre il numero e le dimensioni dei server in base alle esigenze. Questo avviene mediante l'aumento o la riduzione delle risorse nel cloud. Questa capacità di modificare i piani a causa delle fluttuazioni nelle dimensioni e nelle esigenze aziendali è un notevole vantaggio del cloud computing, specialmente quando si verifica una crescita improvvisa della domanda.

2. **Istantaneità:** Tutto ciò di cui hai bisogno è immediatamente disponibile nel cloud.

3. **Risparmio di denaro:** Un vantaggio del cloud computing è la riduzione dei costi dell'hardware. Invece di acquistare attrezzature in loco, le esigenze hardware vengono lasciate al fornitore. Per le entità che crescono rapidamente, il nuovo hardware può essere grande, costoso e scomodo. Il cloud computing allevia questi problemi perché le risorse possono essere acquisite in modo rapido e semplice. Inoltre, il costo della riparazione o sostituzione dell'attrezzatura viene trasferito ai fornitori. Oltre ai costi di acquisto, l'hardware esterno riduce i costi interni di energia e risparmia spazio. I grandi data center possono occupare spazio d'ufficio prezioso e produrre una grande quantità di calore. Passare a applicazioni o archiviazione basate su cloud può contribuire a massimizzare lo spazio e ridurre significativamente le spese energetiche.

4. **Affidabilità:** Invece di essere ospitato su una singola istanza di un server fisico, l'hosting viene fornito su una partizione virtuale che trae le sue risorse, come lo spazio su disco, da una vasta rete di server fisici sottostanti. Se un server va offline, ciò non avrà alcun effetto sulla disponibilità, poiché i server virtuali continueranno a prelevare risorse dalla rete rimanente.

5. **Sicurezza fisica:** I server fisici sottostanti sono comunque ospitati nei data center e ciò contribuisce a beneficiare delle misure di sicurezza che tali strutture implementano per impedire alle persone di accedervi o di disturbare le operazioni in loco.

6. **Esternalizzazione della gestione:** Quando gestisci la tua organizzazione, qualcun altro gestisce l'infrastruttura informatica. Non devi preoccuparti della gestione e degli aggiornamenti.

## Cloud Providers

## 

Il cloud computing può essere definito come la pratica di utilizzare una rete di server remoti ospitati su Internet per archiviare, gestire ed elaborare dati, anziché utilizzare un server locale o un computer personale. Le aziende che offrono servizi di cloud computing di questo tipo sono chiamate fornitori di servizi cloud e generalmente addebitano i servizi di cloud computing in base all'uso. Griglie e cluster sono le fondamenta del cloud computing.

Il **cloud computing** è stato efficace nel cambiare la distribuzione commerciale dei sistemi. Alcuni esempi includono:

1. **Amazon Web Services (AWS):** uno dei business basati su cloud di maggior successo è Amazon Web Services (AWS), che è un'offerta di Infrastructure as a Service (IaaS) che paga l'affitto per computer virtuali sull'infrastruttura di Amazon.

2. **Piattaforma Microsoft Azure:** Microsoft sta creando la piattaforma Azure che consente all'applicazione del framework .NET di funzionare su Internet come piattaforma alternativa per i developer di Microsoft. Questo è il classico Platform as a Service (PaaS).

3. **Google:** Google ha costruito una rete mondiale di data center per servire il suo motore di ricerca. Da questo servizio, Google ha catturato i ricavi mondiali della pubblicità. Utilizzando tali ricavi, Google offre software gratuito agli utenti basato sull'infrastruttura. Questo è chiamato Software as a Service (SaaS).

4. **IBM Cloud** è una raccolta di servizi di cloud computing per le organizzazioni forniti dalla IBM Corporation. Fornisce Infrastructure as a Service, Software as a Service e Platform as a Service.

5. **Oracle Cloud** è una raccolta di servizi cloud offerti da Oracle Corporation, tra cui Infrastructure as a Service (IaaS), Platform as a Service (PaaS) e Software as a Service (SaaS).

6. **Alibaba Cloud** è il braccio di cloud computing del gruppo Alibaba, che fornisce una suite completa di servizi globali di cloud computing per alimentare sia le attività online dei clienti internazionali che l'ecosistema di e-commerce di Alibaba Group.

Questi sono solo alcuni esempi della vasta gamma di applicazioni per il cloud computing. Con l'avanzare della tecnologia, le possibilità del cloud computing continueranno a espandersi.

I servizi di **cloud computing** più comuni rientrano in cinque categorie principali:

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)

Questi sono talvolta chiamati lo stack del cloud computing perché sono costruiti uno sopra l'altro. Conoscere cosa sono e come sono diversi rende più semplice la scelta dei servizi più appropriati per gli utenti. Questi livelli di astrazione possono anche essere visti come un'architettura stratificata in cui i servizi di un livello superiore possono essere composti da servizi del livello sottostante, ad esempio SaaS può fornire l'infrastruttura.

Software as a Service (SaaS) è una modalità di erogare servizi e applicazioni via Internet. Invece di installare e gestire software, è possibile accedervi via Internet, liberandosi dalla gestione complessa di software e hardware. La maggior parte delle applicazioni SaaS può essere eseguita direttamente da un browser web senza necessità di download o installazioni.

**Vantaggi di SaaS:**

1. **Economico:** si paga solo per ciò che si utilizza.

2. **Tempo ridotto:** gli utenti possono eseguire la maggior parte delle applicazioni SaaS direttamente dal proprio browser web senza dover scaricare e installare alcun software.

3. **Accessibilità:** è possibile accedere ai dati dell'applicazione ovunque.

4. **Aggiornamenti automatici:** anziché acquistare nuovo software, i clienti si affidano al provider SaaS per eseguire automaticamente gli aggiornamenti.

5. **Scalabilità:** consente agli utenti di accedere ai servizi e alle funzionalità su richiesta.

Alcune delle diverse aziende che forniscono Software as a Service includono Cloud9 Analytics, Salesforce.com, Cloud Switch, Microsoft Office 365, Big Commerce, Eloqua, Dropbox e Cloud Tran.

**Svantaggi di PaaS:**

**Controllo limitato sull'infrastruttura:** i fornitori PaaS gestiscono tipicamente l'infrastruttura sottostante e si occupano della manutenzione e degli aggiornamenti, ma ciò può anche significare che gli utenti hanno meno controllo sull'ambiente e potrebbero non essere in grado di apportare determinate personalizzazioni.

**Dipendenza dal provider:** gli utenti dipendono dal fornitore PaaS per la disponibilità, la scalabilità e l'affidabilità della piattaforma, il che può rappresentare un rischio se il fornitore si trova ad affrontare interruzioni o altri problemi.

**Flessibilità limitata**: le soluzioni PaaS potrebbero non essere in grado di gestire determinati tipi di carichi di lavoro o applicazioni, il che può limitare il valore della soluzione per alcune organizzazioni.

Ci sono altri servizi come "Infrastructure as a Service" (IaaS), "Hardware as a Service" (HaaS), "Function as a Service" (FaaS) che possono essere appresi da diverse fonti (non inclusi in questo documento).

## La virtualizzazione nel cloud computing e i tipi

## La virtualizzazione

La virtualizzazione è una tecnica che riguarda il modo in cui separare un servizio dalla distribuzione fisica sottostante di quel servizio. Si tratta del processo di creazione di una versione virtuale di qualcosa, come l'hardware del computer. È stata sviluppata inizialmente durante l'era dei mainframe. Coinvolge l'uso di software specializzati per creare una versione virtuale o creata tramite software di una risorsa informatica invece della versione fisica reale della stessa risorsa. Grazie alla virtualizzazione, è possibile eseguire contemporaneamente più sistemi operativi e applicazioni sulla stessa macchina fisica e sullo stesso hardware, aumentando l'utilizzo e la flessibilità dell'hardware.

In altre parole, una delle principali tecniche economiche, di riduzione dell'hardware e di risparmio energetico utilizzate dai fornitori di servizi cloud è la virtualizzazione. La virtualizzazione consente di condividere una singola istanza fisica di una risorsa o di un'applicazione tra più clienti e organizzazioni contemporaneamente. Ciò avviene assegnando un nome logico all'archiviazione fisica e fornendo un puntatore a tale risorsa fisica su richiesta. Il termine virtualizzazione è spesso sinonimo di virtualizzazione dell'hardware, che svolge un ruolo fondamentale nel fornire in modo efficiente soluzioni di Infrastructure-as-a-Service (IaaS) per il cloud computing. Inoltre, le tecnologie di virtualizzazione forniscono un ambiente virtuale non solo per l'esecuzione di applicazioni, ma anche per lo storage, la memoria e la rete.

**Ruolo della virtualizzazione nel cloud computing:**

La virtualizzazione ha un impatto significativo sul cloud computing. Nel caso del cloud computing, gli utenti archiviano dati nel cloud, ma grazie alla virtualizzazione hanno il vantaggio aggiuntivo di condividere l'infrastruttura. I fornitori di servizi cloud si occupano delle risorse fisiche necessarie, ma questi provider di servizi cloud addebitano una cifra considerevole per questi servizi, il che influisce su ogni utente o organizzazione. La virtualizzazione aiuta gli utenti o le organizzazioni a mantenere quei servizi necessari a un'azienda attraverso persone esterne (terze parti), riducendo i costi aziendali. Questo è il modo in cui funziona la virtualizzazione nel cloud computing.

**Vantaggi della virtualizzazione:**

1. Assegnazione più flessibile ed efficiente delle risorse.
2. Aumento della produttività nello sviluppo.
3. Riduzione dei costi dell'infrastruttura IT.
4. Accesso remoto e scalabilità rapida.
5. Alta disponibilità e ripristino da disastro.
6. Pagamento per l'uso dell'infrastruttura IT su richiesta.
7. Consente l'esecuzione di più sistemi operativi contemporaneamente.

**Caratteristiche della virtualizzazione:**

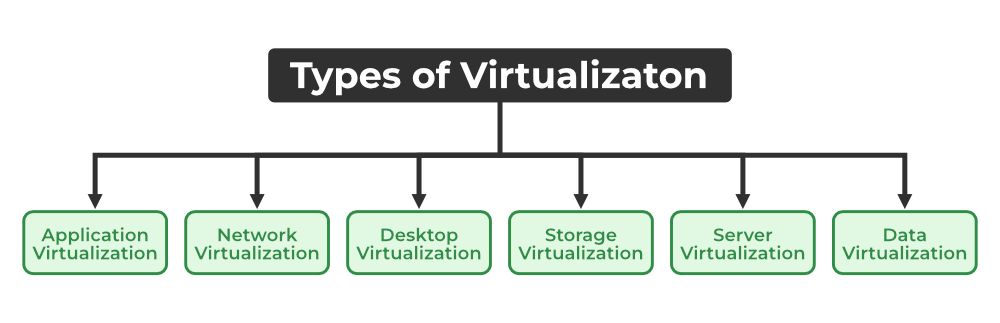
**Sicurezza aumentata:** la capacità di controllare l'esecuzione di un programma ospite in modo completamente trasparente apre nuove possibilità per la distribuzione di un ambiente di esecuzione sicuro e controllato. Tutte le operazioni dei programmi ospiti vengono generalmente eseguite sulla macchina virtuale, che le traduce e le applica ai programmi host.

**Esecuzione gestita:** in particolare, condivisione, aggregazione, emulazione e isolamento sono le caratteristiche più rilevanti.

**Condivisione:** la virtualizzazione consente la creazione di un ambiente di elaborazione separato all'interno dello stesso host.

**Tipi di virtualizzazione:**

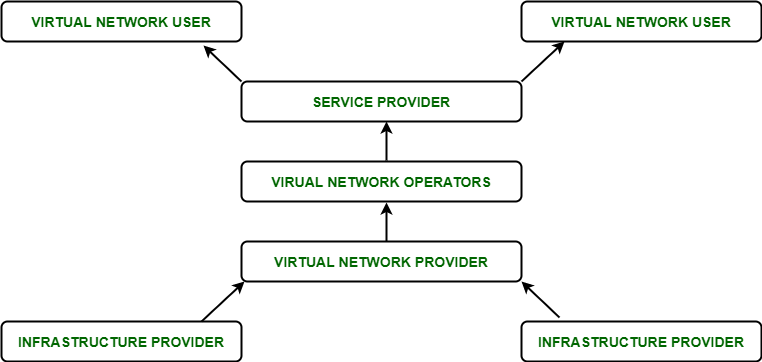
1. Virtualizzazione dell'applicazione.
2. Virtualizzazione di rete.
3. Virtualizzazione del desktop.
4. Virtualizzazione dello storage.
5. Virtualizzazione del server.
6. Virtualizzazione dei dati.

****

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

1. Virtualizzazione delle applicazioni: La virtualizzazione delle applicazioni consente a un utente di avere l'accesso remoto a un'applicazione da un server. Il server archivia tutte le informazioni personali e le altre caratteristiche dell'applicazione, ma può comunque essere eseguito su una postazione di lavoro locale tramite Internet. Un esempio di questo sarebbe un utente che ha bisogno di eseguire due diverse versioni dello stesso software. Le tecnologie che utilizzano la virtualizzazione delle applicazioni includono applicazioni ospitate e applicazioni confezionate.

2. Virtualizzazione di rete: La capacità di eseguire più reti virtuali, ciascuna con un piano di controllo e dati separato. Esse coesistono insieme su una singola rete fisica. Possono essere gestite da parti individuali che sono potenzialmente riservate l'una all'altra. La virtualizzazione di rete fornisce la possibilità di creare e predisporre reti virtuali, switch logici, router, [firewalls](https://www.geeksforgeeks.org/introduction-of-firewall-in-computer-network/), bilanciatori di carico, reti private virtuali [VPN](https://www.geeksforgeeks.org/virtual-private-network-vpn-introduction/)) e sicurezza del carico di lavoro in giorni o addirittura settimane ([www.geeksforgeek.org](http://www.geeksforgeek.org)). [9]



Fonte: www.Cloudtech.com

3. Virtualizzazione desktop: La virtualizzazione desktop consente di memorizzare in remoto il sistema operativo degli utenti su un server nel data center. Questo permette all'utente di accedere virtualmente al proprio desktop da qualsiasi posizione e da una macchina diversa. Gli utenti che desiderano sistemi operativi specifici diversi da Windows Server avranno bisogno di un desktop virtuale. I principali vantaggi della virtualizzazione desktop sono la mobilità dell'utente, la portabilità e la gestione semplificata dell'installazione del software, degli aggiornamenti e dei patch.

4. Virtualizzazione dello storage: La virtualizzazione dello storage è una serie di server gestiti da un sistema di storage virtuale. I server non sono consapevoli di dove esattamente sono memorizzati i loro dati e funzionano più come api operai in un alveare. Questo consente di gestire lo storage da diverse fonti come se fosse un unico repository. Il software di virtualizzazione dello storage mantiene operazioni fluide, prestazioni consistenti e una serie continua di funzioni avanzate nonostante cambiamenti, guasti e differenze nell'hardware sottostante.

5. Virtualizzazione del server: Questo tipo di virtualizzazione comporta la mascheratura delle risorse del server. Qui, il server centrale (server fisico) viene suddiviso in più server virtuali diversi mediante la modifica del numero di identificazione e dei processori. Quindi, ogni sistema può operare i propri sistemi operativi in modo isolato. Ogni sottoserver conosce l'identità del server centrale. Questo aumenta le prestazioni e riduce i costi operativi mediante la distribuzione delle risorse del server principale in risorse sottoserver. È vantaggioso per la migrazione virtuale, la riduzione del consumo energetico, la riduzione dei costi infrastrutturali, ecc.

6. Virtualizzazione dei dati: Si tratta di una forma di virtualizzazione in cui i dati vengono raccolti da varie fonti e gestiti in un unico luogo, senza la necessità di conoscere dettagli tecnici come il modo in cui i dati vengono raccolti, memorizzati e formattati. Questi dati vengono quindi organizzati in modo logico in modo che la loro vista virtuale possa essere accessibile da persone interessate e stakeholder attraverso vari servizi cloud in remoto. Numerose grandi aziende forniscono servizi di questo tipo, come Oracle, IBM, AtScale, CData, ecc.

Usi della virtualizzazione:

* Integrazione dei dati
* Integrazione aziendale
* Servizi dati architettura orientata ai servizi
* Ricerca di dati organizzativi
* Ricerca di dati organizzativi

**Conclusioni**

L'informatica in cloud offre grandi opportunità per organizzazioni, aziende e individui, come elencato in questa sezione. Questa tecnologia ha attirato l'attenzione sia dell'industria pubblica che privata negli ultimi dieci anni. Tuttavia, è possibile affermare che non è ancora riconosciuta e utilizzata appieno. C'è ancora un lungo cammino da percorrere, specialmente considerando il crescente bisogno di memorizzazione dei dati online in tutto il mondo. Inoltre, osservando i vantaggi che l'informatica in cloud può offrire, si presume che sarà ampiamente adottata da persone e organizzazioni in un prossimo futuro, con investimenti concomitanti da parte delle industrie. Questa previsione è supportata dalla crescente quantità di fornitori di servizi cloud che offrono piattaforme, strumenti e servizi cloud.

# 

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

# 

# 

# 

# 

# 

# Deployment

## Secure DevOps

DevOps è un insieme di tecniche che enfatizza la collaborazione e l'automazione tra i team di sviluppo e operazioni. DevSecOps è l'integrazione delle pratiche di sicurezza in questa metodologia. DevSecOps cerca di stabilire una cultura della sicurezza che garantisce che il software sia sicuro e rispetti gli standard di conformità integrando la sicurezza in ogni fase del ciclo di sviluppo del software, dalla pianificazione alla distribuzione.

**Come funziona DevSecOps?**

DevSecOps è fondamentalmente basato sulla cooperazione e la comunicazione tra i team di sviluppo, sicurezza e operazioni. Ciò implica che tutti abbiano una qualche responsabilità per la sicurezza, non solo il team di sicurezza. Invece di aggiungere funzionalità di sicurezza in seguito, i team di sviluppo collaborano con i team di sicurezza e operazioni per incorporare la sicurezza nel software fin dall'inizio.

L'automazione di DevSecOps è un elemento cruciale. L'automazione rende il processo di sviluppo più efficiente, riduce gli errori e garantisce la coerenza. DevSecOps può aiutare a individuare vulnerabilità e minacce in modo più rapido e preciso mediante l'automatizzazione dei test di sicurezza e altre operazioni di sicurezza.

**Quali sono i vantaggi di DevSecOps?**

Uno dei principali vantaggi di DevSecOps è la capacità di identificare e affrontare i rischi di sicurezza in fase di sviluppo. Ciò significa che la sicurezza è incorporata nel software fin dall'inizio, anziché essere aggiunta successivamente, il che può essere costoso e richiedere molto tempo. Inoltre, DevSecOps svolge un ruolo importante nella riduzione del rischio di violazioni di sicurezza e perdite di dati mediante l'identificazione precoce delle vulnerabilità.

Un altro aspetto cruciale della pratica è che aiuta a garantire la conformità alle leggi e agli standard. In molte aziende, specialmente in settori che trattano dati sensibili o privati, come il settore sanitario e bancario, la conformità sta diventando sempre più importante. DevSecOps contribuisce a garantire che il software sia conforme ai requisiti integrando la conformità nel processo di sviluppo.

## Provisioning e Deprovisioning

**Cos'è il Provisioning e il Deprovisioning?**

Il provisioning è il processo di mettere a disposizione dei sistemi di tecnologia dell'informazione (IT) per gli utenti. A seconda delle esigenze dell'organizzazione, il provisioning può riguardare la rete, i server, le applicazioni e gli utenti:

Il provisioning di rete comporta la creazione di una rete a cui possono accedere utenti, server e dispositivi. Ad esempio, l'industria delle telecomunicazioni utilizza il provisioning di rete per fornire soluzioni wireless ai clienti.

Il provisioning del server è il processo di configurazione di un server che può essere utilizzato all'interno di una rete. Questo può includere la creazione di una nuova macchina, l'installazione e la configurazione del software e la connessione alle reti e allo storage.

Il provisioning delle applicazioni è una soluzione di gestione delle infrastrutture che consente agli amministratori di ottimizzare le prestazioni per vari ambienti all'interno di un'azienda.

Il provisioning degli utenti è il processo di gestione delle identità digitali, che include la creazione, l'aggiornamento e la rimozione di diritti e autorizzazioni per le applicazioni, i file, le reti, i sistemi e le risorse aziendali.

Il deprovisioning è il processo di rimozione dell'accesso dell'utente ai servizi software e di rete. In breve, è l'opposto del provisioning ed è tipicamente necessario quando i dipendenti cambiano ruolo o lasciano l'azienda.

Tanto il provisioning quanto il deprovisioning svolgono un ruolo importante nella sicurezza dei sistemi IT e delle applicazioni. Tuttavia, un provisioning automatico ed efficace dovrebbe essere una priorità per qualsiasi organizzazione desideri migliorare la sua postura di sicurezza.

**Perché il provisioning e il deprovisioning degli utenti sono importanti**

Quando un nuovo dipendente viene assunto, una delle prime cose che un'organizzazione fa è creare un record di quel dipendente. Successivamente, spetta al reparto delle risorse umane, all'IT o a una combinazione di entrambi, fornire a quel dipendente l'accesso a tutte le app, gli account e i sistemi di cui ha bisogno per svolgere il proprio lavoro.

La fornitura degli utenti avviene ogni volta che vengono aggiunte o modificate informazioni nei sistemi HR dell'organizzazione: ciò include l'aggiunta di membri del team, cambiamenti di ruolo, promozioni e trasferimenti tra i dipartimenti, tra le altre cose. In altre parole, la fornitura degli utenti ti aiuta a fornire il giusto livello di accesso agli utenti corretti durante l'ingresso in azienda, a mantenere aggiornato l'accesso durante il periodo di impiego e, nel processo di deprovisioning, a rimuovere l'accesso quando un dipendente lascia l'organizzazione.

**Cos'è la fornitura automatizzata e quali sono i suoi vantaggi?**

La fornitura automatizzata significa rendere automatici i processi manuali di ammissione e uscita degli utenti. In organizzazioni di ogni dimensione, la fornitura automatizzata degli utenti libera il personale IT e HR per lavorare su compiti più strategici, previene lacune nella sicurezza minimizzando l'impatto degli errori umani e offre una migliore esperienza agli utenti.

L'aggiornamento manuale dei profili degli utenti, dei privilegi dell'account e delle appartenenze ai gruppi richiede tempo, specialmente quando i dipendenti hanno bisogno di accesso a un numero sempre maggiore di applicazioni aziendali. Inoltre, il processo può subire ritardi se i team IT sono occupati con altri progetti. Ciò può impedire una rapida ammissione dei nuovi utenti, rallentare l'assegnazione o la revoca dei diritti di accesso e rendere più difficile il monitoraggio e l'individuazione di permessi non necessari. La fornitura e la deprovisioning automatizzati alleviano questa pressione sul personale IT, consentendo a tutti di dedicare il proprio tempo a progetti che creano valore aziendale.

Automatizzando la fornitura degli utenti, è possibile eliminare alcune delle lacune nella gestione dell'accesso che potrebbero rendere l'azienda vulnerabile a violazioni della sicurezza. La creazione manuale di account utente significa che qualcuno all'interno dell'organizzazione deve condividere una password con un dipendente, il che è probabilmente un processo molto insicuro, ad esempio inviando una e-mail o scrivendo su un promemoria. Situazioni simili di errori umani possono rappresentare una minaccia alla sicurezza. Ad esempio, gli utenti potrebbero essere accidentalmente forniti di accesso a sistemi e dati a cui non dovrebbero avere accesso, o potrebbero ancora avere accesso una volta che lasciano l'organizzazione.

L'automatizzazione della fornitura e deprovisioning degli utenti elimina questi rischi, fornendo agli individui autorizzazioni in modo sicuro e privato. Il processo garantisce che un utente venga fornito di accesso alle applicazioni in loco ed esterne in base agli attributi del proprio ruolo. Questi attributi e permessi vengono quindi memorizzati in un'unica posizione centrale, garantendo che possano essere facilmente modificati quando il ruolo di un dipendente cambia. Quando i dipartimenti o i team implementano nuovi strumenti o modificano i diritti dei dipendenti, l'accesso può essere anche distribuito in base a regole di gruppo.

La fornitura automatizzata degli utenti fornisce l'accesso solo quando è necessario, prevenendo eventuali falle nella sicurezza che i pirati informatici potrebbero sfruttare per ottenere l'accesso non autorizzato a informazioni aziendali sensibili.

## 

## Tecniche di Secure Coding

**Cos'è la codifica sicura?**

La codifica sicura, anche conosciuta come programmazione sicura, consiste nel scrivere codice in un linguaggio ad alto livello che segue rigorosi principi, con l'obiettivo di prevenire potenziali vulnerabilità (che potrebbero esporre dati o causare danni all'interno di un sistema mirato). La codifica sicura va oltre la semplice scrittura, compilazione e rilascio del codice nelle applicazioni. Per abbracciare appieno la programmazione sicura, è necessario creare un ambiente di sviluppo sicuro costruito su un'infrastruttura IT affidabile e sicura utilizzando hardware, software, servizi e fornitori sicuri.

**Perché è importante la codifica sicura?**

Sempre più transazioni finanziarie si stanno spostando online. Gli incidenti di sicurezza spesso hanno origine nel software sottostante di un'applicazione e possono avere gravi conseguenze per le aziende e gli individui. Codice non sicuro in settori importanti (ad esempio finanza, sanità, energia e trasporti) potrebbe causare danni finanziari e materiali, manipolazioni di mercato e furto, persino danni fisici e morti.

Il pericolo è reale: nei recenti anni, i report dei media hanno messo in luce quanto sia insicuro gran parte del software che utilizziamo. Anche le grandi organizzazioni con risorse e conoscenze a disposizione hanno subito gravi violazioni dei dati. Per le aziende che forniscono software ai consumatori o alle imprese, la fiducia dei clienti è estremamente preziosa, e perderla potrebbe influire sul loro bilancio. Pertanto, garantire pratiche di codifica sicura deve essere una priorità assoluta per queste organizzazioni.

**5 migliori pratiche per la codifica sicura:**

**Minificazione e offuscamento del codice:** Rendere il codice più difficile da accedere e, di conseguenza, più difficile da leggere, può scoraggiare potenziali attacchi. Nel mondo di JavaScript, una pratica comune è la minificazione del codice. La minificazione rimuove gli spazi vuoti e le interruzioni di riga dal codice, riducendo la dimensione dei file di codice e rendendo il codice esposto molto più difficile da leggere. Un'altra tecnica simile e più efficace è l'offuscamento del codice, che trasforma il codice leggibile dall'essere umano in testo difficile da capire.

**Evitare scorciatoie:** Può essere allettante per gli sviluppatori voler prendere scorciatoie per rilasciare il codice più velocemente, ma ciò potrebbe comportare gravi implicazioni per la sicurezza. Ad esempio, gli attacchi spesso avvengono quando le credenziali rigide e i token di sicurezza sono lasciati come commenti. Queste informazioni dovrebbero essere rimosse molto prima del rilascio delle applicazioni.

**Scansioni automatizzate e revisioni del codice:** Gli attacchi come l'iniezione di script tra siti (XSS), l'iniezione di SQL e altri tipi di attacchi possono sfruttare le vulnerabilità di sicurezza nel codice. Una combinazione di regolari revisioni del codice sicuro e strumenti automatizzati che eseguono scansioni sul codice può contribuire a prevenire tali attacchi.

**Evitare componenti con vulnerabilità conosciute:** Anche se i componenti e le librerie open source possono risparmiare tempo ed energie agli sviluppatori, sono anche un comune punto di ingresso per attacchi malevoli e una grande fonte di vulnerabilità ed exploit potenziali. Evitare l'uso di componenti con vulnerabilità conosciute e monitorare costantemente nuove vulnerabilità durante il processo di sviluppo aiuterà a mantenere l'integrità del codice.

**Audit e registrazione:** Un software con registrazione e monitoraggio sufficienti consente di individuare potenziali incidenti quando il codice è implementato in un ambiente di produzione.

**Cos'è il controllo di versione?**

Il controllo di versione, noto anche come controllo del sorgente, è la pratica di tracciare e gestire le modifiche al codice software. I sistemi di controllo di versione sono strumenti software che aiutano i team di sviluppo a gestire le modifiche al codice sorgente nel tempo. Con l'accelerazione degli ambienti di sviluppo, i sistemi di controllo di versione aiutano i team di sviluppo software a lavorare più velocemente ed in modo più intelligente. Sono particolarmente utili per i team di DevOps poiché li aiutano a ridurre i tempi di sviluppo e aumentare le distribuzioni di successo.

Il software di controllo di versione tiene traccia di ogni modifica al codice in un tipo speciale di database. Se viene commesso un errore, gli sviluppatori possono tornare indietro nel tempo e confrontare le versioni precedenti del codice per aiutare a correggere l'errore minimizzando le interruzioni per tutti i membri del team. Per la maggior parte dei progetti software, il codice sorgente è come le gioielli della corona: un prezioso asset il cui valore deve essere protetto. Per la maggior parte dei team di sviluppo software, il codice sorgente è un repository della conoscenza e comprensione inestimabili del dominio del problema che gli sviluppatori hanno raccolto e affinato attraverso un attento lavoro. Il controllo di versione protegge il codice sorgente da catastrofi e dalla degradazione casuale degli errori umani e delle conseguenze non intenzionali.

I developer di software che lavorano in team scrivono continuamente nuovo codice sorgente e apportano modifiche al codice sorgente esistente. Il codice per un progetto, un'applicazione o un componente software è solitamente organizzato in una struttura a cartelle o "albero dei file". Un membro del team può essere al lavoro su una nuova funzionalità, mentre un altro sviluppatore corregge un bug non correlato cambiando il codice; ogni sviluppatore può apportare le proprie modifiche in diverse parti dell'albero dei file.

Il controllo di versione aiuta i team a risolvere questo tipo di problemi, tracciando ogni singola modifica apportata da ciascun contributore e aiutando a prevenire il conflitto tra il lavoro concorrente. Le modifiche apportate in una parte del software possono essere incompatibili con quelle apportate da un altro sviluppatore che lavora nello stesso momento. Questo problema dovrebbe essere individuato e risolto in modo ordinato senza bloccare il lavoro degli altri membri del team. Inoltre, in tutti gli sviluppi software, qualsiasi modifica può introdurre nuovi bug e il nuovo software non può essere considerato affidabile finché non è stato testato. Quindi, il testing e lo sviluppo procedono insieme fino a quando non è pronta una nuova versione.

**Cos'è la gestione dei cambiamenti?**

La gestione dei cambiamenti IT è un processo strutturato per valutare i cambiamenti proposti a sistemi o servizi IT. Questa procedura viene eseguita prima di implementare il cambio richiesto sulla rete di un'organizzazione, riducendo o eliminando le interruzioni di rete.

La gestione dei cambiamenti IT è necessaria per garantire che eventuali modifiche alla rete non degradino le prestazioni della rete stessa. Le modifiche alla rete dovrebbero essere azioni definite e intenzionali per eliminare una vulnerabilità scoperta, migliorare le prestazioni di un componente di rete o sostituire un componente di rete obsoleto o difettoso.

**Quali sono i tipi di cambiamenti IT?**

I cambiamenti dei sistemi o servizi IT sono suddivisi in tre tipi:

1. **Cambiamenti Standard:** Questi cambiamenti sono routine e seguono un processo predefinito per l'analisi dei rischi e le preapprovazioni. Sono processi già convalidati e preapprovati per l'esecuzione. Esempi di cambiamenti standard includono l'aggiornamento della RAM o delle dimensioni del disco rigido, la sostituzione di un dispositivo di rete guasto o la creazione di una nuova istanza del database.

2. **Cambiamenti Normali:** Questi cambiamenti non seguono un processo predefinito. È necessario presentare un'analisi dei rischi e un piano di implementazione per ottenere l'approvazione prima di attuare questi cambiamenti nella rete IT. Esempi di cambiamenti normali includono l'aggiornamento a un nuovo sistema di gestione della conformità, l'aggiornamento dei dispositivi di rete per migliorare le prestazioni o la ricollocazione di una farm di server.

3. **Cambiamenti d'Emergenza:** Questi cambiamenti avvengono quando si verifica o è probabile un'interruzione non pianificata a causa di una vulnerabilità scoperta che rappresenta una minaccia significativa per la rete. Esempi di cambiamenti d'emergenza includono l'installazione di un patch di sicurezza, un'interruzione di un dispositivo di rete o il ripristino da un incidente grave (ad esempio un taglio di fibra).

## Code Quality & Security

**Cos'è la qualità del codice?**

Quando cerchiamo la parola "qualità" nel dizionario, essa è definita come "quanto buono o cattivo è qualcosa". Pertanto, quando parliamo di "qualità del codice", ci riferiamo a una metrica per valutare quanto buono o cattivo sia un insieme di istruzioni software. Ma quanto è buono o cattivo in base a cosa? In generale, in base a una varietà di caratteristiche o attributi. Tuttavia, come può accadere, ad esempio, con la classificazione del comportamento umano da una prospettiva morale, la qualità del codice non è una metrica definitiva con parametri sempre condivisi ed oggettivi ed è aperta a discussioni. Dipende dalla soggettività, da ciò che le industrie e le organizzazioni definiscono in base alle loro esigenze specifiche, ai requisiti e agli approcci. Ad esempio, la qualità del codice non sarà necessariamente vista allo stesso modo da coloro che sviluppano semplici giochi per dispositivi mobili e da coloro che sviluppano programmi che controllano le macchine di enormi aziende elettriche, entrambi software con criticità così diverse.

Tuttavia, alcune idee specifiche sembrano essere condivise per quasi tutti i progetti software che consentono di separare il codice di buona qualità da quello di cattiva qualità. Il codice di cattiva qualità può mancare di coerenza e coerenza nella gestione delle convenzioni ed essere pieno di bug e complessità. Il codice di buona qualità, al contrario, è di solito considerato semplice, privo di bug, ben documentato e adempiente alla sua funzione prevista per gli utenti finali. Di seguito sono riportate alcune delle proprietà chiave condivise per valutare la qualità del codice:

* **Affidabilità:** misura quanto è probabile che il software funzioni senza errori, raggiungendo i suoi scopi durante un periodo specifico. Questa proprietà dipende dal numero di errori presenti nel codice.
* **Robustezza:** misura quanto bene il software può affrontare comportamenti utente strani e altre condizioni utilizzando messaggi di errore comprensibili. Questa proprietà è legata alla ridotta suscettibilità a bug nascosti o all'introduzione di nuovi bug.
* **Testabilità:** misura quanto bene il software supporta l'uso di test che possono essere impiegati, ad esempio, per verificare determinati comportamenti o rilevare guasti nella sua funzionalità.
* **Leggibilità:** misura quanto sia leggibile e comprensibile il codice del software non solo per i suoi autori originali, ma anche per coloro che intendono esaminarlo e modificarlo. Dipende dall'uso di commenti, notazioni, rientri, documentazione e semplicità, tra le altre cose.
* **Manutenibilità:** misura la facilità con cui il software può essere mantenuto, cioè riparato, aggiornato e migliorato. Questa proprietà dipende dalla struttura del codice, dalle dimensioni, dalla coerenza e dalla complessità.
* **Portabilità:** misura quanto il software è utilizzabile su diversi dispositivi, piattaforme o altri ambienti. In altre parole, misura quanto sia facilmente trasferibile da uno all'altro, a seconda del numero di modifiche richieste.
* **Riutilizzabilità:** misura quanto le parti del codice o gli asset del software possono essere replicati o riutilizzati (anche per costruirvi sopra) da sviluppatori in altri progetti o programmi.

Fino a questo punto, ciò che è problematico è che dopo aver esaminato il web e consultato diverse fonti, in generale non troviamo spesso la sicurezza o almeno non esplicitamente suggerita tra queste proprietà chiave.

**E la sicurezza?**

La qualità del codice è solitamente associata alle prestazioni del codice e all'esperienza degli utenti finali e degli sviluppatori che ci lavorano. Tuttavia, la qualità del codice non dovrebbe essere misurata anche in base alla sua sicurezza? D'altra parte, la sicurezza non dipende anche da ciò che chiamiamo qualità? Quasi un decennio fa, un gruppo di ricercatori condivise le seguenti parole in uno studio per l'Institute di Ingegneria del Software:

Molte delle Enumerazioni delle Debolezze Comuni (CWE), come l'uso improprio delle costruzioni del linguaggio di programmazione, i buffer overflow e le mancate convalide dei valori di input, possono essere associate a pratiche di sviluppo e codifica di scarsa qualità. Migliorare la qualità è una condizione necessaria per affrontare alcune problematiche relative alla sicurezza del software.

Pezzi di codice di scarsa qualità, anche con errori minori, risultanti da pratiche di codifica inadeguate, generano debolezze di sicurezza e vulnerabilità che possono essere sfruttate da hacker malintenzionati e generare impatti negativi significativi su organizzazioni e utenti finali. La sicurezza dipende quindi dalla qualità del prodotto. Ma si tratta di una qualità diversa che non riguarda solo le proprietà elencate in precedenza. In questa "qualità del codice migliorata" vengono introdotti standard e pratiche che, se non rispettate, ci portano a parlare di insicurezza, proprio come la non conformità ad altri requisiti potrebbe portare a parlare di, ad esempio, mancanza di affidabilità o scarsa leggibilità. Introducendo questi nuovi standard, abbiamo incluso la sicurezza come proprietà nella lista precedente. Pertanto, d'ora in poi, nel nostro discorso, la sicurezza del codice verrà considerata come un fattore che gioca un ruolo nella determinazione della qualità del codice.

## 

## 

Le vulnerabilità di sicurezza nel codice mettono a rischio non solo le preziose prestazioni del software, ma anche la privacy dei dati sensibili nel caso vengano elaborati lì. La mancanza di inclusione della sicurezza nel concetto di qualità del codice spesso visto potrebbe essere in parte correlata a quest'ultimo. In passato, a differenza dei giorni nostri, i prodotti software che non coinvolgevano l'uso di dati sensibili dell'organizzazione e dei suoi utenti e clienti o il contatto con minacce esterne erano più abbondanti. In quel periodo, c'era una maggiore preoccupazione per proprietà come l'affidabilità, la testabilità e la manutenibilità. Oggi, tuttavia, nella nostra economia, c'è molta più dipendenza da Internet e da applicazioni web e mobili che trasferiscono dati sensibili, nonché un numero maggiore di dispositivi IoT e interconnessioni. Questo è il motivo per cui è sempre più necessario che la sicurezza appaia anche come una proprietà chiave in relazione alla qualità del codice.

## 

## OSWAP

## 

## 

## 

## 

## 

## 

**Cos'è OWASP?**

Open Web Application Security Project, o OWASP, è un'organizzazione internazionale no-profit dedicata alla sicurezza delle applicazioni web. Uno dei principi fondamentali di OWASP è che tutti i loro materiali siano liberamente disponibili ed accessibili sul loro sito web, permettendo a chiunque di migliorare la propria sicurezza delle applicazioni web. I materiali offerti includono documentazione, strumenti, video e forum. Forse il loro progetto più noto è l'OWASP Top 10.

**Cos'è l'OWASP Top 10?**

L'OWASP Top 10 è un rapporto regolarmente aggiornato che elenca le preoccupazioni sulla sicurezza delle applicazioni web, concentrandosi sui 10 rischi più critici. Il rapporto è redatto da un team di esperti di sicurezza provenienti da tutto il mondo. OWASP definisce il Top 10 come un "documento di sensibilizzazione" e raccomanda a tutte le aziende di incorporare il rapporto nei loro processi al fine di ridurre e/o mitigare i rischi per la sicurezza.

Di seguito sono elencati i rischi per la sicurezza riportati nel rapporto dell'OWASP Top 10:

1. **Iniezione**

Gli attacchi di iniezione avvengono quando dati non fidati vengono inviati a un interprete di codice tramite un input di modulo o qualche altra sottomissione di dati a un'applicazione web. Ad esempio, un attaccante potrebbe inserire del codice SQL in un modulo che si aspetta un nome utente in testo normale. Se quell'input del modulo non è adeguatamente protetto, ciò comporterebbe l'esecuzione di quel codice SQL. Questo è noto come un attacco di iniezione SQL.

Gli attacchi di iniezione possono essere prevenuti con la convalida e/o la sanificazione dei dati inviati dagli utenti. (La convalida significa rifiutare dati dallo sguardo sospetto, mentre la sanificazione si riferisce alla pulizia delle parti sospette dei dati.) Inoltre, un amministratore del database può impostare controlli per ridurre al minimo la quantità di informazioni che un attacco di iniezione può esporre.

1. **Autenticazione difettosa**

Le vulnerabilità nei sistemi di autenticazione (accesso) possono concedere agli attaccanti l'accesso agli account degli utenti e persino la capacità di compromettere l'intero sistema utilizzando un account amministratore. Ad esempio, un attaccante può prendere una lista contenente migliaia di combinazioni di nomi utente/password noti ottenuti durante una violazione dei dati e utilizzare uno script per provare tutte queste combinazioni su un sistema di accesso per vedere se ce ne sono che funzionano.

Alcune strategie per mitigare le vulnerabilità di autenticazione includono l'obbligo dell'autenticazione a due fattori (2FA) nonché il limitare o ritardare i tentativi ripetuti di accesso mediante il controllo della frequenza.

1. **Esposizione di dati sensibili**

Se le applicazioni web non proteggono dati sensibili come informazioni finanziarie e password, gli attaccanti possono accedere a tali dati e venderli o utilizzarli per scopi maliziosi. Un metodo popolare per rubare informazioni sensibili è l'utilizzo di un attacco "on-path".

Il rischio di esposizione dei dati può essere ridotto crittografando tutti i dati sensibili e disabilitando la memorizzazione nella cache\* di eventuali informazioni sensibili. Inoltre, gli sviluppatori di applicazioni web dovrebbero fare attenzione a non conservare dati sensibili in modo non necessario.

\*La memorizzazione nella cache è la pratica di conservare temporaneamente i dati per il riutilizzo. Ad esempio, i browser web memorizzano spesso le pagine web in modo che, se un utente ritorna su quelle pagine entro un determinato intervallo di tempo, il browser non debba scaricare nuovamente le pagine dal web.

1. **Entità esterne XML (XEE)**

Questo è un attacco contro un'applicazione web che analizza l'input XML\*. Questo input può fare riferimento a un'entità esterna, cercando di sfruttare una vulnerabilità nel parser. Una "entità esterna" in questo contesto si riferisce a un'unità di archiviazione, come un disco rigido. Un parser XML può essere ingannato nel trasmettere dati a un'entità esterna non autorizzata, che può passare dati sensibili direttamente a un attaccante.

I modi migliori per prevenire gli attacchi XEE sono quelli di far accettare alle applicazioni web un tipo di dati meno complesso, come JSON\*\*, o almeno di applicare patch ai parser XML e disabilitare l'uso di entità esterne in un'applicazione XML.

\*XML o Extensible Markup Language è un linguaggio di markup progettato per essere leggibile sia da esseri umani che da macchine. A causa della sua complessità e delle vulnerabilità di sicurezza, è in via di smantellamento in molte applicazioni web.

\*\*JavaScript Object Notation (JSON) è un tipo di notazione semplice e leggibile da esseri umani spesso utilizzata per trasmettere dati su Internet. Anche se è stato originariamente creato per JavaScript, JSON è linguaggio-agnostico e può essere interpretato da molti diversi linguaggi di programmazione.

1. **Violazione del Controllo di Accesso**

Il controllo di accesso si riferisce a un sistema che controlla l'accesso alle informazioni o alle funzionalità. La violazione del controllo di accesso consente agli aggressori di eludere l'autorizzazione e di eseguire operazioni come se fossero utenti privilegiati, come gli amministratori. Ad esempio, un'applicazione web potrebbe consentire a un utente di cambiare l'account con cui ha effettuato l'accesso semplicemente modificando una parte dell'URL, senza alcuna altra verifica.

I controlli di accesso possono essere resi sicuri assicurandosi che un'applicazione web utilizzi dei token di autorizzazione e stabilisca rigidi controlli su di essi.

\*Molti servizi rilasciano token di autorizzazione quando gli utenti effettuano l'accesso. Ogni richiesta privilegiata effettuata dall'utente richiederà che il token di autorizzazione sia presente. Questo è un modo sicuro per garantire che l'utente sia chi dice di essere, senza dover costantemente inserire le proprie credenziali di accesso.

1. **Configurazione di sicurezza difettosa**

La configurazione di sicurezza difettosa è la vulnerabilità più comune nella lista ed è spesso il risultato dell'uso di configurazioni predefinite o di errori eccessivamente dettagliati. Ad esempio, un'applicazione potrebbe mostrare errori eccessivamente descrittivi all'utente che potrebbero rivelare vulnerabilità nell'applicazione. Questo può essere mitigato rimuovendo le funzionalità non utilizzate nel codice e assicurandosi che i messaggi di errore siano più generici.

1. **Cross-Site Scripting (XSS)**

Le vulnerabilità di cross-site scripting si verificano quando le applicazioni web consentono agli utenti di inserire codice personalizzato in un percorso URL o su un sito web che verrà visto da altri utenti. Questa vulnerabilità può essere sfruttata per eseguire codice JavaScript maligno nel browser di una vittima. Ad esempio, un attaccante potrebbe inviare un'email a una vittima che sembra provenire da una banca di fiducia, con un collegamento al sito web di quella banca. Questo collegamento potrebbe contenere del codice JavaScript maligno aggiunto alla fine dell'URL. Se il sito della banca non è adeguatamente protetto contro il cross-site scripting, allora quel codice maligno verrà eseguito nel browser della vittima quando fa clic sul collegamento.

Le strategie di mitigazione per il cross-site scripting includono l'elaborazione delle richieste HTTP non attendibili, nonché la convalida e/o la sanificazione dei contenuti generati dagli utenti. L'uso di moderni framework di sviluppo web come ReactJS e Ruby on Rails fornisce anche una certa protezione integrata contro il cross-site scripting.

1. **Deserializzazione non sicura**

Questa minaccia prende di mira molte applicazioni web che frequentemente serializzano e deserializzano dati. La serializzazione significa prendere oggetti dal codice dell'applicazione e convertirli in un formato che può essere utilizzato per altri scopi, come archiviare i dati su disco o trasmetterli in streaming. La deserializzazione è l'opposto: convertire i dati serializzati in oggetti che l'applicazione può utilizzare. La serializzazione è in qualche modo simile a imballare i mobili in scatole prima di un trasloco, mentre la deserializzazione è come sballare le scatole e assemblare i mobili dopo il trasloco. Un attacco di deserializzazione non sicura è come se i traslocatori manomettessero il contenuto delle scatole prima che vengano sballate.

Uno sfruttamento di deserializzazione non sicura è il risultato della deserializzazione di dati da fonti non attendibili e può comportare conseguenze gravi come attacchi DDoS e attacchi di esecuzione di codice a distanza. Sebbene siano state adottate misure per cercare di individuare gli attaccanti, come il monitoraggio della deserializzazione e l'implementazione di controlli sui tipi, il modo più sicuro per proteggersi dagli attacchi di deserializzazione non sicura è vietare la deserializzazione dei dati provenienti da fonti non attendibili.

1. **Utilizzo di componenti con vulnerabilità conosciute**

Molti sviluppatori web moderni utilizzano componenti come librerie e framework nelle loro applicazioni web. Questi componenti sono pezzi di software che aiutano gli sviluppatori a evitare lavoro ridondante e a fornire funzionalità necessarie. Esempi comuni includono framework front-end come React e librerie più piccole utilizzate per aggiungere icone di condivisione o test A/B. Alcuni attaccanti cercano vulnerabilità in questi componenti che poi possono essere utilizzate per orchestrare attacchi. Alcuni dei componenti più popolari vengono utilizzati su centinaia di migliaia di siti web; un attaccante che trova una falla di sicurezza in uno di questi componenti potrebbe lasciare centinaia di migliaia di siti vulnerabili agli attacchi.

I produttori di componenti spesso offrono patch di sicurezza e aggiornamenti per chiudere le vulnerabilità conosciute, ma gli sviluppatori di applicazioni web non hanno sempre le versioni patchate o più recenti dei componenti in esecuzione sulle loro applicazioni. Per ridurre al minimo il rischio di eseguire componenti con vulnerabilità conosciute, gli sviluppatori dovrebbero rimuovere i componenti non utilizzati dai loro progetti e assicurarsi di ricevere componenti da una fonte attendibile e che siano aggiornati.

1. **Registrazione e monitoraggio insufficienti**

Molte applicazioni web non intraprendono sufficienti misure per rilevare le violazioni dei dati. Il tempo medio di scoperta di una violazione è di circa 200 giorni dopo che è avvenuta. Questo dà agli attaccanti molto tempo per causare danni prima di ricevere una risposta. OWASP raccomanda che gli sviluppatori web implementino registrazione e monitoraggio, nonché piani di risposta agli incidenti, per garantire di essere informati degli attacchi alle loro applicazioni.

## Conclusioni

Lo sviluppo di applicazioni sicure affronta direttamente l'integrazione della sicurezza in ogni fase del ciclo di vita dello sviluppo del software, dalla pianificazione alla distribuzione. Questa è un'area operativa altamente necessaria che funziona per stabilire una cultura della sicurezza che garantisce che il software sia sicuro e conforme agli standard di conformità. Allo stesso modo, la fornitura automatica significa rendere automatici i processi manuali di inserimento e cancellazione degli utenti. In organizzazioni di tutte le dimensioni, la fornitura automatica degli utenti libera l'IT e le risorse umane per lavorare su compiti più strategici, evita falle nella sicurezza riducendo l'impatto degli errori umani e offre migliori esperienze agli utenti. Inoltre, la scrittura di codice sicuro, anche definita programmazione sicura, comporta la scrittura di codice in un linguaggio ad alto livello che segue rigorosi principi, con l'obiettivo di prevenire potenziali vulnerabilità che potrebbero esporre dati o causare danni all'interno di un sistema mirato.

# 

# 3[.3 Autemticazione & Aut](https://docs.google.com/document/d/1wCWqOdq8zWOGG18ngM5w-qUsIoc4DANG/edit#heading=h.g98km9urykdd)orizzazione

# 

**Metodi di Autenticazione**

**Protocollo di Autenticazione con Password (PAP)**

Il PAP, o protocollo di autenticazione con password, è un metodo di autenticazione del protocollo punto-punto (PPP) che utilizza le password per convalidare gli utenti. È uno standard Internet (RFC 1334), un protocollo di autenticazione basato su password.

Utilizzando il PAP, i dati non sono crittografati. Sono inviati al server di autenticazione come testo normale. Il PAP utilizza una procedura di autenticazione a due vie per autenticare gli utenti in base all'username e alla password forniti.

Quando utilizzato in PPP, il protocollo di autenticazione con password è considerato un sistema di autenticazione debole. Poiché i dati non sono crittografati, sono vulnerabili e visibili a un attore malintenzionato in grado di visualizzare la sessione PPP.

L'uso di CHAP (protocollo di autenticazione challenge-handshake) può aggiungere uno strato aggiuntivo di sicurezza alla sessione PPP mediante l'aggiunta di un processo di autenticazione a tre vie. Il PAP è una procedura di accesso standard utilizzata come metodo PPP per l'autenticazione degli utenti.

**Token di Autenticazione**

L'autenticazione basata su token è un protocollo che consente agli utenti di verificare la propria identità e ricevere un token di accesso univoco in cambio. Durante la durata del token, gli utenti possono quindi accedere al sito web o all'applicazione per cui è stato emesso il token, senza dover reinserire le credenziali ogni volta che tornano alla stessa pagina web, app o risorsa protetta dallo stesso token.

I token di autenticazione funzionano come biglietti timbrati. L'utente mantiene l'accesso finché il token rimane valido. Una volta che l'utente effettua il logout o chiude un'app, il token viene invalidato.

L'autenticazione basata su token è diversa dalle tecniche di autenticazione tradizionali basate su password o server. I token offrono un secondo livello di sicurezza, e gli amministratori hanno un controllo dettagliato su ciascuna azione e transazione.

Tuttavia, l'uso dei token richiede una certa conoscenza di codifica. La maggior parte degli sviluppatori impara rapidamente le tecniche, ma c'è una curva di apprendimento.

## 

## 

## 

## 

## 

## 

## 

## 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]

**Autenticazione tramite Token in 4 Semplici Passaggi**

Utilizzare un sistema di autenticazione basato su token consente ai visitatori di verificare le proprie credenziali solo una volta. In cambio, riceveranno un token che consente l'accesso per un periodo di tempo da te definito. Il processo funziona nel seguente modo:

1. **Richiesta:** La persona richiede l'accesso a un server o a una risorsa protetta. Questa richiesta può implicare un login con una password o un altro processo da te specificato.

2. **Verifica:** Il server determina se la persona dovrebbe avere accesso. Ciò può comportare il controllo della password rispetto all'username o un altro processo da te specificato.

3. **Token:** Il server comunica con il dispositivo di autenticazione, come un anello, una chiave, un telefono o dispositivo simile. Dopo la verifica, il server emette un token e lo passa all'utente.

4. **Conservazione:** Il token rimane nel browser dell'utente mentre l'operazione continua.

Se l'utente tenta di accedere a una parte diversa del server, il token comunica nuovamente con il server. L'accesso viene concesso o negato in base al token.

Gli amministratori impostano limiti sui token. È possibile consentire un token monouso che viene immediatamente distrutto quando l'utente effettua il logout. Oppure è possibile impostare il token in modo che si auto-distrugga alla fine di un periodo di tempo specificato.

**Autenticazione con Chiave Simmetrica**

Una chiave simmetrica è una chiave che può essere utilizzata sia per cifrare che per decifrare i dati. Ciò significa che per decifrare i dati, è necessario avere la stessa chiave utilizzata per cifrarli. La crittografia simmetrica è generalmente più efficace della crittografia asimmetrica e pertanto è preferita quando è necessario scambiare grandi quantità di dati.

La creazione della chiave condivisa può essere complessa utilizzando solo algoritmi di crittografia simmetrica, quindi in alcuni casi viene utilizzata la crittografia asimmetrica per creare la chiave condivisa tra due parti.

L'algoritmo Standard di Crittografia Digitale (DES) è un approccio di crittografia a chiave simmetrica comunemente utilizzato nei sistemi di smart card. Questo approccio richiede una chiave crittografica segreta memorizzata e l'algoritmo DES pubblico in ogni smart card e CAD. Le varie fasi dell'autenticazione con chiave simmetrica sono le seguenti:

1. La smart card invia il numero di serie del microprocessore (I) al CAD, che collega il numero alla chiave principale (Mk) per formare la chiave diversificata della smart card (K). L'emittente carica una chiave diversificata in ogni smart card durante l'inizializzazione della carta.

2. Il CAD genera un numero casuale (R), quindi cifra R per formare il valore Y, che viene inviato come sfida alla smart card.

3. La smart card decifra Y, formando la risposta (X) e la restituisce al CAD.

4. Il CAD confronta R e X, accettando la carta se i due valori coincidono.

L'algoritmo Telepass 1 è un algoritmo utilizzato per l'autenticazione delle smart card. Questo algoritmo richiede una chiave segreta diversificata, il contenuto di una parola specifica nella memoria della smart card e un valore esterno casuale per calcolare la risposta a una sfida di autenticazione. L'algoritmo Telepass 1 supporta la funzionalità di segretezza dei dati nella distribuzione delle chiavi e l'integrità dei dati tramite codici di autenticazione dei messaggi.

In un sistema a chiave simmetrica, la chiave segreta in ogni smart card deve essere unica in modo che la scoperta della chiave non comprometta l'intero sistema. Questa diversificazione della chiave supporta una chiave crittografica univoca per ogni smart card durante la personalizzazione. Il sistema crea la chiave diversificata a partire dal set di una chiave principale di sistema e una caratteristica della scheda unica, inclusa il numero di serie del microprocessore.

Il CAD include una copia della chiave principale in modo da poter generare la chiave modificata per ciascuna smart card. La compromissione di una chiave diversificata ha un impatto limitato sul sistema, a condizione che la compromissione venga rilevata e che la carta possa essere rimossa dal sistema. La compromissione di una chiave principale può avere gravi conseguenze.

La compromissione di una chiave principale richiede che tutte le carte siano caricate con nuove chiavi, il che è costoso per i sistemi e scomodo per gli utenti. Per evitare la compromissione delle chiavi, il CAD può includere un modulo di sicurezza con memoria protetta (ad esempio, una smart card) in cui è memorizzata la chiave principale.

## Autenticazione Biometrica

**Cos'è l'Autenticazione Biometrica?**

L'autenticazione è un modo per verificare, senza ombra di dubbio, che una persona sia chi dice di essere. L'autenticazione biometrica esegue questa verifica controllando le caratteristiche biologiche o comportamentali distintive.[5]

Un sistema di autenticazione funziona confrontando i dati forniti con le informazioni utente convalidate memorizzate in un database. Nei sistemi tradizionali, queste informazioni sono le password. Nell'autenticazione biometrica, queste informazioni sono definite come tratti fisici o comportamentali.

Ad esempio, in un sistema di riconoscimento facciale, diverse caratteristiche facciali vengono elaborate e convertite in dati numerici, che vengono memorizzati in un database. Quando una persona cerca di effettuare l'accesso, il sistema cattura nuovamente il loro volto, estrae i dati numerici e li confronta con ciò che è memorizzato nel database. Altri tipi di autenticazione biometrica includono:

* Scansione delle impronte digitali
* Corrispondenza del DNA
* Scansione della retina
* Scansione delle vene

## Biometria comportamentale

La biometria comportamentale verifica l'identità analizzando il comportamento fisico e cognitivo di un utente. Utilizza algoritmi di machine learning per determinare modelli nel comportamento e nelle attività dell'utente. Questi modelli vengono quindi utilizzati per rilevare se qualcuno è chi dice di essere.

Esempi di biometria comportamentale includono:

* Uso dello schermo touchscreen (quanta area dello schermo stanno utilizzando)
* Dinamiche di scrittura (scorciatoie da tastiera o velocità di scrittura)
* Attività del mouse

## Autorizzazione

L'autorizzazione è la procedura che consente a qualcuno di fare qualcosa. Definisce un approccio per verificare se l'utente ha il permesso di accedere a una risorsa o meno. Può rappresentare quali dati e informazioni un utente può accedere.[11]

Viene anche chiamata AuthZ. L'autorizzazione funziona generalmente con l'autenticazione in modo che il sistema possa comprendere chi sta accedendo alle informazioni. L'autorizzazione è una struttura di sicurezza utilizzata per determinare i privilegi dell'utente/cliente o i livelli di accesso associati alle risorse di sistema, come programmi informatici, file, servizi, dati e funzionalità dell'applicazione.

Di solito, l'autorizzazione è preceduta dall'autenticazione per la verifica dell'identità del cliente. Gli amministratori di sistema (SA) vengono generalmente assegnati ai livelli di autorizzazione che coprono alcune risorse di sistema e clienti.[40]

Durante l'autorizzazione, un sistema verifica le regole di accesso di un utente autenticato e concede o nega l'accesso alle risorse. I moderni sistemi operativi multiutente basati su processi di autorizzazione progettati in modo efficiente per supportare l'implementazione e la gestione delle applicazioni.

Fattori chiave come il tipo di utente, il numero e le credenziali che richiedono verifica e le azioni e i ruoli associati. Ad esempio, l'autorizzazione basata su ruoli può essere assegnata dai gruppi di utenti che necessitano di privilegi di tracciamento delle risorse specifiche dell'utente.

Inoltre, l'autorizzazione può basarsi su una struttura di autenticazione aziendale, come Active Directory (AD), per un'integrazione delle politiche di sicurezza senza soluzione di continuità. Ad esempio, ASP.NET funziona con Internet Information Server (IIS) e Microsoft Windows per supportare servizi di autenticazione e autorizzazione per applicazioni .NET basate su Internet.[40]

Windows utilizza il File System con Nuova Tecnologia (NTFS) per supportare le Liste di Controllo di Accesso (ACL) per alcune risorse. L'ACL funge da autorità finale sull'accesso alle risorse. Il framework .NET supporta un metodo di sicurezza basato su ruoli alternativi per il supporto dell'autorizzazione.

## API Keys

**Cosa sono le chiavi API?**

"Che cos'è una chiave API?". Le interfacce di programmazione delle applicazioni (API) consentono ai programmi software di interagire, condividere dati ed integrare le loro funzionalità. Le API facilitano le conversazioni tra software altrimenti sconnessi e sono la tecnologia alla base di molte potenti integrazioni che si usano quotidianamente. [30]

Tuttavia, a differenza di una conversazione faccia a faccia, è più difficile per un'API verificare se l'app con cui sta comunicando è davvero chi afferma di essere. Poiché spesso espongono informazioni private e sensibili, le API hanno bisogno di metodi per identificare i loro clienti prima di consentire l'accesso. Altrimenti, si tratterebbe di un rischio significativo per la sicurezza.

Le chiavi API sono uno di questi metodi di sicurezza. Agiscono come una sorta di carta d'identità per il cliente che effettua una richiesta API, aiutando le API a assegnare le adeguate autorizzazioni di accesso e a tenere traccia di come vengono utilizzati i dati.

## Basic Auth

**Cos'è l'Autenticazione Base?**

Ampiamente utilizzato nelle comunicazioni basate su HTTP, il metodo base è il mezzo per autenticare gli utenti finali prima di concedere l'accesso alle risorse o alla comunicazione. A tal fine, agli utenti previsti viene chiesto di fornire credenziali primarie come nome utente e password di accesso.

In questo metodo, i dati codificati in base64 vengono trasmessi tramite un'intestazione di autorizzazione.

Poiché non richiede l'uso di cookie, pagine di accesso e altri identificatori, è considerato la tecnica di autenticazione utente più austera che contribuisce a ottenere un controllo completo dell'accesso. [10]

## HMAC

HMAC (Hash-based Message Authentication Code) è un tipo di codice di autenticazione di messaggio (MAC) ottenuto eseguendo una funzione hash crittografica sui dati da autenticare e una chiave segreta condivisa. Come qualsiasi altro MAC, è utilizzato sia per l'integrità dei dati che per l'autenticazione. Verificare l'integrità dei dati è necessario per le parti coinvolte nella comunicazione. Protocolli di trasferimento come HTTPS, SFTP, FTPS e altri utilizzano HMAC. La funzione hash crittografica può essere MD-5, SHA-1 o SHA-256. Le firme digitali sono quasi simili agli HMAC, poiché entrambe utilizzano una funzione hash e una chiave condivisa. La differenza sta nelle chiavi: gli HMAC utilizzano chiavi simmetriche (copie identiche), mentre le firme utilizzano chiavi asimmetriche (due chiavi diverse).

## OAuth

OAuth è un protocollo o framework di autorizzazione open standard che descrive come server e servizi non correlati possono consentire in modo sicuro l'accesso autenticato alle loro risorse senza condividere effettivamente le credenziali di accesso iniziali correlate. Nel linguaggio dell'autenticazione, questo è noto come autorizzazione delegata sicura di terze parti tramite un agente utente.

**Conclusioni**

I metodi di autenticazione e autorizzazione svolgono ruoli importanti nel garantire la sicurezza di varie operazioni digitali che coinvolgono direttamente le persone, come ad esempio coloro che utilizzano strumenti digitali. Essi aiutano gli strumenti digitali a identificare le vere identità delle persone (come utenti) e possono essere utilizzati sia per crittografare che per decrittografare dati, tra le altre cose. Queste azioni consentono alle piattaforme di identificare le persone reali che hanno accesso ufficiale o legittimo a dati privati, che possono essere raggiunti solo dalle persone autorizzate. Uno dei metodi di autenticazione è l'autenticazione biometrica, che verifica le persone controllando caratteristiche biologiche o comportamentali distinte.

L'autorizzazione è il procedimento che permette a qualcuno di fare qualcosa. Essa rappresenta un approccio per verificare se l'utente ha il permesso di accedere a una risorsa o meno. Inoltre, essa determina quali dati e informazioni un utente può accedere. Durante l'autorizzazione, un sistema verifica le regole di accesso di un utente autenticato e concede o nega l'accesso alle risorse. Considerando i moderni sistemi operativi multiutente delle organizzazioni odierna, è necessario progettare processi di autorizzazione efficienti per supportare la distribuzione e l'amministrazione delle applicazioni.

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

## Ridondanza

La ridondanza è un termine ingegneristico che significa "la duplicazione di componenti o funzioni critiche di un sistema con l'intenzione di aumentarne la affidabilità, di solito sotto forma di backup o fail-safe, o per migliorare le prestazioni effettive del sistema".

Nell'informatica, ci sono quattro forme principali di ridondanza:

* Hardware
* Software
* Informazione
* Tempo

Quando parliamo di introdurre ridondanza in un progetto infrastrutturale, ci riferiremo molto probabilmente alla ridondanza hardware. La ridondanza hardware significa semplicemente aggiungere un dispositivo o componente duplicato all'interno del sistema. Questo interviene quando un dispositivo o componente principale fallisce. L'obiettivo è garantire zero tempo di inattività.

**Ridondanza del server**

In alcune aziende, l'ambiente informatico include la ridondanza del server. Per abilitare questa funzionalità, viene creata una replica del server con la stessa potenza di calcolo, archiviazione, applicazioni e altri parametri operativi.

Un server ridondante viene mantenuto offline, cioè è acceso con connettività di rete/Internet ma non viene utilizzato come "server attivo". In caso di guasto, tempo di inattività o traffico eccessivo sul server principale, è possibile attivare un server ridondante per sostituire il server principale o condividere il suo carico.

**Ridondanza di rete**

La ridondanza di rete è un processo mediante il quale vengono installate istanze aggiuntive o alternative di dispositivi di rete, apparecchiature o mezzi di comunicazione all'interno dell'infrastruttura di rete. Si tratta di un metodo per garantire la disponibilità di rete in caso di guasto di un dispositivo di rete o di un percorso di comunicazione e la conseguente indisponibilità. Pertanto, fornisce un mezzo di failover di rete.

La ridondanza di rete viene principalmente implementata nell'infrastruttura di rete aziendale per fornire una fonte ridondante di comunicazioni di rete. Essa funge da meccanismo di backup per passare rapidamente le operazioni di rete su un'infrastruttura ridondante in caso di interruzioni di rete non pianificate.

**Ridondanza di Internet**

Una forma di ridondanza di rete è il Multi-WAN, disponibile in alcuni firewall come i firewall UTM WatchGuard forniti da Your IT. Il Multi-WAN consente di avere più connessioni Internet in un singolo sito. Se una connessione Internet si interrompe, è possibile passare a un'altra, garantendo una connessione continua.

## Tipi di Backup

**3 Principali Tipi di Strategie di Backup dei Dati**

Esaminiamo i vantaggi e gli svantaggi dei principali metodi di backup dei dati: backup completo, differenziale e incrementale, per assicurarsi di scegliere la strategia di backup dei dati ottimale per le esigenze della vostra organizzazione.

1. **Backup Completi dei Dati**

Come suggerisce il nome, un backup completo dei dati si riferisce a una copia completa di tutti i dati, inclusi file e cartelle, indipendentemente da backup precedenti o da circostanze specifiche. L'intero set di dati viene copiato e memorizzato in loco, in un'altra posizione o in entrambi. I backup completi sono i più rapidi da ripristinare, poiché tutti i file sono contenuti nello stesso set di backup.

I backup completi di solito forniscono la migliore protezione contro la perdita critica dei dati, ma spesso vengono effettuati periodicamente secondo un programma predefinito a causa del tempo e dei costi coinvolti. Tuttavia, qualsiasi piano di backup valido dovrebbe includere almeno un backup completo dei dati, almeno una volta, come base.

Un backup completo dei dati è generalmente il più semplice da eseguire ma non necessariamente il più semplice da amministrare. Ecco alcuni svantaggi da considerare:

Utilizza la maggior parte dello spazio di archiviazione tra tutti i metodi

Richiede una larghezza di banda di rete significativa

Spesso richiede hardware aggiuntivo se il set di dati è grande

Richiede un tempo di implementazione considerevole, a seconda delle dimensioni dell'azienda

Gli altri due tipi di backup, differenziale e incrementale, derivano da un backup completo dei dati e lavorano con esso. I backup differenziali e incrementali sono simili e presentano alcune differenze sottili ma importanti.

1. **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]

1. **Backup Incrementale dei Dati**

Similmente ai backup differenziali, i backup incrementali iniziano con un backup completo per memorizzare tutti i file. Successivamente, i backup incrementali archiviano solo le aggiunte e le modifiche apportate al backup precedente o all'ultimo backup incrementale più recente. Mentre un backup differenziale contiene tutti i dati dall'ultimo backup completo, un backup incrementale include solo i dati che sono cambiati rispetto a qualsiasi backup precedente.

Poiché i backup incrementali memorizzano solo le modifiche (incrementi), richiedono meno spazio di archiviazione, consentendo velocità di backup accelerate. Tuttavia, hanno il processo di ripristino più lento. Per ripristinare un sistema completo, un'azienda deve prima ripristinare l'ultimo backup completo e quindi ripristinare ciascun backup incrementale in ordine. Il ripristino completo diventa sempre più difficile o impossibile se un backup incrementale viene perso, danneggiato o rubato.

## Conclusioni

Le strategie di implementazione della resilienza informatica sono argomenti importanti per proteggere le organizzazioni da vari attacchi informatici, come la perdita di dati, reputazione, risorse, denaro e fiducia, dove le implementazioni di resilienza informatica aiutano tali organizzazioni a rimanere al sicuro il più possibile attraverso la ridondanza del server, la ridondanza di rete e la ridondanza di Internet e le strategie di backup dei dati.

# 3.5 Securing Embedded Systems

La sicurezza dei sistemi embedded è una strategia che fornisce protezione ai sistemi embedded da attacchi informatici. I sistemi embedded, noti anche come "computer embedded", sono piccoli dispositivi con una funzione dedicata all'interno di un sistema più ampio.

I dispositivi Internet delle cose (IoT) sono esempi di sistemi embedded. Ad esempio, elettrodomestici intelligenti come il frigorifero, il termostato e l'allarme di sicurezza svolgono funzioni diverse. Tuttavia, fanno parte del sistema domestico complessivo e sono interconnessi tra loro tramite Internet. La sicurezza dei sistemi embedded mira a proteggere il software in esecuzione su questi dispositivi, poiché sempre più attori malintenzionati li prendono di mira.

## Come funziona la sicurezza dei sistemi embedded?

Sebbene la sicurezza dei sistemi embedded sia una strategia, essa comprende diversi sistemi di sicurezza e strumenti. Esistono diversi fornitori di sicurezza dei sistemi embedded. Ad esempio, Kaspersky Embedded Systems Security protegge i sistemi embedded Windows. Star Lab è un altro fornitore specializzato nella protezione dei sistemi embedded nei settori aerospaziale e militare.

Indipendentemente dall'azienda o dal fornitore, la sicurezza dei sistemi embedded presenta aspetti cruciali, tra cui:

* Monitoraggio delle minacce: la sicurezza dei sistemi embedded dovrebbe monitorare continuamente tutte le parti del sistema embedded per eventuali minacce potenziali. Una patch dovrebbe essere progettata e distribuita immediatamente quando viene rilevata una minaccia.
* Miglioramento continuo del design della sicurezza: la sicurezza dei sistemi embedded lavora incessantemente sul design della sicurezza all'evolversi dei nuovi metodi di attacco. Anche quando la tattica di attacco informatico non è ancora rilevata nel sistema embedded, il design della sicurezza deve essere aggiornato per proteggersi dalle nuove tattiche maliziose.
* Aggiornamenti regolari della sicurezza: è fondamentale aggiornare frequentemente il software dei sistemi embedded per proteggere i dispositivi dalle minacce emergenti.
* Supporto 24x7: i fornitori di sicurezza dei sistemi embedded dovrebbero offrire supporto 24 ore su 24 per aiutare i team interni con l'integrazione, i test e altri processi coinvolti nelle operazioni del sistema embedded.

## 

**IoT**

I dispositivi IoT e ICS sono considerati punti finali. In altre parole, sono dispositivi situati alla fine di una catena di comunicazione che inizia con una persona o un dispositivo robotico e termina nelle piattaforme cloud e nei data center. I dispositivi IoT e ICS non appaiono dal nulla. Sono progettati, sviluppati e gestiti, proprio come qualsiasi altro computer. La Figura 1 fornisce una panoramica degli elementi all'interno di un tipico dispositivo IoT.

Ogni dispositivo IoT include ciascuno di questi elementi:

1. Firmware: Memoria a sola lettura incorporata nel dispositivo che fornisce un controllo a basso livello dell'hardware. Può essere aggiornata, ma di solito non programmata. Comunica tra ciascuno degli elementi nel dispositivo e altri dispositivi in rete.

2. Servizi di protezione: Una parte del firmware o del sistema operativo del dispositivo che fornisce funzionalità di sicurezza, compresa la capacità di isolare i processi in modo che non possano essere utilizzati per eludere la sicurezza e l'encryption.

3. Sensori di movimento: Hardware e software combinati utilizzati per monitorare il movimento del dispositivo. Possono includere la rilevazione di movimenti semplici (ad esempio, spostare il dispositivo avanti e indietro o su e giù) o la connettività satellitare (ad esempio, Global Positioning System o GLONASS).

4. Microcontrollore: Il processore utilizzato per eseguire il software e fornire il "cervello" dell'unità.

5. Pila di connettività: Responsabile della connettività di rete. Le capacità di rete possono includere Bluetooth, mobile (ad esempio, 3G, 4G, 5G), Zigbee, LoRA, SigFox o WiFi.

6. Servizi di autenticazione: Quando inclusi, forniscono la capacità per il dispositivo IoT di verificare e convalidare gli utenti, il traffico di rete o i processi.

7. Gestione dell'alimentazione: Se il dispositivo richiede un uso significativo di energia, funzionalità che gestisce l'uso dell'energia, nonché la ricarica del dispositivo.

8. Batteria/Alimentazione: La capacità fisica di immagazzinare energia, nonché di ricevere energia da una fonte remota.

9. Memoria: Il "muscolo" di lavoro del dispositivo IoT, in quanto fornisce la capacità di archiviare dati di lavoro, codice macchina e informazioni che sono poi gestite dal processore.

10. Archiviazione: Fornisce la capacità di catturare e conservare dati per periodi di tempo relativamente lunghi. Tali informazioni possono includere la memorizzazione delle posizioni in cui l'utente del dispositivo IoT ha utilizzato il dispositivo, informazioni su altri dispositivi che si sono connessi ad esso e informazioni inserite dall'utente. Tali informazioni possono essere aggiunte attivamente (ad esempio, dall'utente che programma il dispositivo IoT/OT a scopo deliberato) o passivamente (ad esempio, il dispositivo che cattura i movimenti e le azioni dell'utente).

## Conclusioni

La sicurezza dei sistemi embedded è una strategia che fornisce protezione ai sistemi embedded dai cyber attacchi. I sistemi embedded, noti anche come "computer embedded", sono piccoli dispositivi con una funzione dedicata all'interno di un sistema più ampio che forniscono servizi di sicurezza 24/7.

Inoltre, i dispositivi IoT e ICS sono considerati punti finali. In altre parole, sono dispositivi situati alla fine di una catena di comunicazione che inizia con una persona o un dispositivo robotico e termina nelle piattaforme cloud e nei data center. Sia i dispositivi IoT che i dispositivi ICS sono progettati, sviluppati e gestiti, proprio come qualsiasi altro computer, con vari componenti tra cui firmware, microcontrollore, sensori di movimento, alimentazione, memoria, archiviazione, ecc.

# 3.6 Concetti Base di Crittografia

## Firme Digitali

Una firma digitale è una tecnica matematica che convalida l'autenticità e l'integrità di un messaggio, di un software o di documenti digitali. Permette di verificare il nome dell'autore, la data e l'ora delle firme e autenticare il contenuto del messaggio. La firma digitale offre una sicurezza intrinseca molto maggiore ed è progettata per risolvere il problema della manipolazione e dell'usurpazione (intenzionale copia delle caratteristiche di un'altra persona) nelle comunicazioni digitali.

L'autenticazione basata su informazioni aziendali informatiche interrela sia la tecnologia che la legge. Richiede anche la collaborazione tra persone di diverse esperienze professionali e aree di competenza. Le firme digitali sono diverse da altre firme elettroniche non solo in termini di processo e risultato, ma rendono le firme digitali più utilizzabili per scopi legali. Alcune firme elettroniche che sono legalmente riconoscibili come firme potrebbero non essere sicure come le firme digitali e potrebbero causare incertezza e dispute.

**Applicazione della Firma Digitale**

La ragione principale per implementare una firma digitale nella comunicazione è:

* Autenticazione
* Non ripudio
* Integrità

**Autenticazione:** L'autenticazione è un processo che verifica l'identità di un utente che desidera accedere al sistema. Nella firma digitale, l'autenticazione aiuta ad autenticare le fonti dei messaggi.

**Non ripudio:** Il non ripudio significa l'assicurazione di qualcosa che non può essere negato. Garantisce che qualcuno coinvolto in un contratto o in una comunicazione non possa successivamente negare l'autenticità della sua firma su un documento o in un file o l'invio di un messaggio che ha originato.

**Integrità:** L'integrità garantisce che il messaggio sia reale, accurato e protetto da modifiche non autorizzate da parte di utenti durante la trasmissione.

**Scambio di Chiavi**

Lo scambio di chiavi Internet (IKE) è un protocollo ibrido spesso utilizzato a fini di gestione delle chiavi in reti IPSec. È spesso utilizzato come metodo per lo scambio di chiavi di crittografia e/o chiavi di autenticazione attraverso un mezzo non protetto come Internet. In altre parole, l'Internet Key Exchange mira a fornire crittografia sicura per ambienti non sicuri o vulnerabili.

**Quali sono i vantaggi dello scambio di chiavi Internet?**

L'Internet Key Exchange offre numerosi vantaggi aggiuntivi, tra cui la flessibilità. Di seguito puoi trovare alcuni di questi vantaggi:

L'Internet Key Exchange offre la possibilità di cambiare la crittografia durante le sessioni IPsec.

Attraverso l'uso dell'Internet Key Exchange, viene eliminata la necessità di specificare manualmente tutti i parametri di sicurezza IPsec.

L'Internet Key Exchange consente l'autenticazione dinamica dei peer.

Simmetrica ed Asimmetrica

**Simmetrica**

Si tratta del tipo più semplice di crittografia che coinvolge un'unica chiave segreta per cifrare e decifrare le informazioni. La crittografia simmetrica è una tecnica antica e ben nota. Utilizza una chiave segreta che può essere un numero, una parola o una stringa di lettere casuali. Viene combinata con il testo normale di un messaggio per modificarne il contenuto in un modo specifico. Il mittente e il destinatario devono conoscere la chiave segreta utilizzata per cifrare e decifrare tutti i messaggi. Blowfish, AES, RC4, DES, RC5 e RC6 sono esempi di crittografia simmetrica. L'algoritmo simmetrico più ampiamente utilizzato è AES-128, AES-192 e AES-256.

Il principale svantaggio della crittografia a chiave simmetrica è che tutte le parti coinvolte devono scambiare la chiave utilizzata per cifrare i dati prima di poterli decifrare.

**Vantaggi e Svantaggi della Crittografia Simmetrica**

**Vantaggi:**

Più veloce: poiché utilizza una singola chiave per cifratura e decifratura, è più veloce da eseguire.

Verifica dell'identità: utilizza l'autenticazione tramite password come misura di sicurezza per dimostrare l'identità del destinatario.

Facile da eseguire e gestire: gli utenti hanno una sola chiave per cifrare e decifrare, quindi è facile da eseguire e gestire.

**Svantaggi:**

Le possibilità di condividere chiavi di crittografia in modo sicuro sono ridotte; è difficile e complesso condividere le chiavi nella crittografia simmetrica.

La crittografia simmetrica non è così scalabile, in quanto non è adatta a vari utenti.

**Asimmetrica**

La crittografia asimmetrica, nota anche come crittografia a chiave pubblica, è un metodo relativamente nuovo rispetto alla crittografia simmetrica. Utilizza due chiavi per cifrare un testo normale. Le chiavi segrete vengono scambiate su Internet o in una grande rete, garantendo che persone malintenzionate non ne abusino. È importante notare che chiunque abbia una chiave segreta può decifrare il messaggio, motivo per cui la crittografia asimmetrica utilizza due chiavi correlate per garantire una maggiore sicurezza. Una chiave pubblica è resa liberamente disponibile a chiunque voglia inviarti un messaggio, mentre la seconda chiave privata viene mantenuta segreta e conosciuta solo da te.

Un messaggio cifrato con una chiave pubblica può essere decifrato solo utilizzando una chiave privata, e viceversa, un messaggio cifrato con una chiave privata può essere decifrato solo utilizzando una chiave pubblica. La sicurezza della chiave pubblica non è necessaria poiché è pubblicamente disponibile e può essere passata su Internet. La crittografia asimmetrica ha un potere molto maggiore nel garantire la sicurezza delle informazioni trasmesse durante la comunicazione.

**Vantaggi e Svantaggi della Crittografia Asimmetrica**

**Vantaggi:**

La crittografia asimmetrica utilizza due chiavi, una pubblica e una privata, quindi non ci sono problemi nella distribuzione delle chiavi.

Con un paio di chiavi, è possibile comunicare facilmente con più parti, rendendo la crittografia asimmetrica più scalabile nelle reti di grandi dimensioni.

**Svantaggi:**

**Prestazioni:** la crittografia asimmetrica è più lenta rispetto alla crittografia simmetrica.

Difficoltà di comprensione ed esecuzione: la crittografia asimmetrica non è così facile da implementare e gestire a causa delle dimensioni delle chiavi più grandi.

Crittografia Asimmetrica nei Certificati Digitali

Per utilizzare la crittografia asimmetrica, è necessario trovare un modo per scoprire le chiavi pubbliche. Una tecnica tipica consiste nell'uso di certificati digitali in un modello di comunicazione client-server. Un certificato è un pacchetto di informazioni che identifica un utente e un server, contenente informazioni come il nome dell'organizzazione, l'organizzazione che ha emesso il certificato, l'indirizzo email degli utenti e il paese, nonché la chiave pubblica degli utenti.

Quando un server e un client richiedono una comunicazione crittografata sicura, inviano una query sulla rete all'altra parte, che invia una copia del certificato. La chiave pubblica dell'altra parte può essere estratta dal certificato. Un certificato può anche essere utilizzato per identificare in modo univoco il titolare.

SSL/TLS utilizza sia la crittografia asimmetrica che quella simmetrica. Si possono rapidamente esaminare i certificati SSL digitalmente firmati emessi da autorità di certificazione (CA) fidate.

**Differenza tra Crittografia Simmetrica e Asimmetrica**

La crittografia simmetrica utilizza una singola chiave che deve essere condivisa tra le persone che devono ricevere il messaggio, mentre la crittografia asimmetrica utilizza una coppia di chiavi pubblica e privata per cifrare e decifrare i messaggi durante la comunicazione.

La crittografia simmetrica è una tecnica antica, mentre la crittografia asimmetrica è relativamente nuova.

La crittografia asimmetrica è stata introdotta per complementare il problema intrinseco della necessità di condividere la chiave nel modello di crittografia simmetrica, eliminando la necessità di condividere la chiave utilizzando una coppia di chiavi pubbliche e private.

La crittografia asimmetrica richiede relativamente più tempo rispetto alla crittografia simmetrica.

**Crittografia di Trasporto**

La Sicurezza del Livello di Trasporto (TLS) è un tipo di protocollo di crittografia utilizzato per garantire la sicurezza delle comunicazioni su una rete.

I siti web utilizzano il TLS per proteggere tutte le comunicazioni tra i loro server e i browser web. La crittografia è un abilitatore per raggiungere la conformità e la privacy dei dati, requisiti fondamentali dell'ambiente aziendale. Essa aiuta le organizzazioni a mantenere il controllo sui dati, a proteggere le informazioni sensibili e a garantire la sicurezza delle comunicazioni. Con la crittografia, anche se gli utenti non autorizzati riescono ad accedere a file sensibili, non saranno in grado di leggere i dati.

**Cos'è il TLS?**

La Sicurezza del Livello di Trasporto (TLS) è un tipo di protocollo di crittografia utilizzato per garantire la sicurezza delle comunicazioni su una rete. I siti web utilizzano il TLS per proteggere tutte le comunicazioni tra i loro server e i browser web. La configurazione attenta del TLS fornisce anche proprietà aggiuntive di privacy, come garantire che la futura divulgazione delle chiavi di crittografia non possa essere utilizzata per decifrare comunicazioni TLS registrate in passato.

**Come funziona?**

Quando la connessione tra un client e un server è protetta da TLS, presenta le seguenti proprietà:

1) Utilizza le stesse chiavi crittografiche per cifrare e decifrare le informazioni.

2) L'identità delle parti che comunicano può essere autenticata utilizzando la crittografia a chiave pubblica.

3) I messaggi trasmessi tramite la sicurezza TLS includono un controllo di integrità mediante un codice di autenticazione dei messaggi per prevenire la perdita o l'alterazione non rilevata dei dati durante la trasmissione.

## Hashing

**Cos'è l'hashing nella cibersicurezza?**

In informatica e crittografia, una funzione di hash è una procedura deterministica che prende un input (o "messaggio") e restituisce una stringa di caratteri di dimensione fissa, solitamente chiamata "digest," che è univoca per l'input.

Una funzione di hash viene utilizzata in numerosi algoritmi e protocolli di cibersicurezza, come la memorizzazione delle password e le firme digitali. L'hashing viene inoltre utilizzato in strutture dati, come una tabella hash (una struttura dati che archivia dati), per ricerche e inserimenti veloci.

**Scopo dell'Hashing**

Apprendere la risposta alla domanda su cosa sia l'hashing nella cibersicurezza può aiutare un professionista a utilizzare algoritmi di hashing per la cifratura dei dati e la sicurezza delle informazioni. I professionisti della cibersicurezza convertono un grande blocco di dati di input utilizzando l'algoritmo di hashing in una stringa di lunghezza fissa più piccola come risultato finale.

Le aziende desiderano sempre proteggere i propri server dati e sistemi di archiviazione cloud da vulnerabilità software dannose. L'hashing aiuta i professionisti della cibersicurezza a garantire che i dati archiviati su server e sistemi di archiviazione cloud rimangano incomprensibili per gli hacker.

**A cosa serve l'Hashing?**

L'hashing è una funzione unidirezionale che trasforma un file o una stringa di testo in un digest univoco del messaggio. Il valore hash è calcolato da un algoritmo di hashing utilizzando i dati binari di un particolare file. Ora vediamo le diverse applicazioni dell'hashing nella cibersicurezza.

**Archiviazione delle Password**

Gli hash forniscono sicurezza al sistema informatico di un'organizzazione in modo che gli hacker non possano rubare, ad esempio, le password email memorizzate sui server.

**Firme Digitali**

L'hashing è un modo per cifrare e decifrare firme digitali, verificando il mittente e il destinatario del messaggio.

**Gestione Documenti**

L'autenticità dei dati può essere verificata con l'uso di algoritmi di hashing. Quando un documento è completamente redatto, lo specialista della cibersicurezza utilizzerà un hash per proteggerlo.

**Gestione dei File**

Le aziende utilizzano gli hash per indicizzare dati, riconoscere file ed eliminare file duplicati. Un'organizzazione può risparmiare tempo considerevole utilizzando gli hash quando lavora con un sistema informatico contenente migliaia di file.

**Steganografia**

Una tecnica di steganografia implica la nascita di informazioni sensibili all'interno di un file o messaggio ordinario e non segreto, in modo che non venga rilevata. Le informazioni sensibili verranno quindi estratte dal file o messaggio ordinario alla destinazione, evitando così la rilevazione.

La steganografia è un passo aggiuntivo che può essere utilizzato insieme alla crittografia per nascondere o proteggere i dati.

La steganografia è un modo per nascondere informazioni segrete all'interno (o persino sulla superficie) di un documento o di un altro supporto altrimenti comune e non segreto per evitarne la rilevazione. Il termine deriva dalle parole greche "steganos," che significa "coperto" o "nascosto," e "graph," che significa "scrivere". Da qui "scrittura nascosta".

Puoi utilizzare la steganografia per nascondere testo, video, immagini o persino dati audio. È una conoscenza utile, limitata solo dal tipo di supporto e dall'immaginazione dell'autore.

## Conclusioni

I concetti crittografici sono diventati una parte importante della cibersicurezza in cui essi convalidano l'autenticità e l'integrità di un messaggio, software o documenti digitali. Ci permettono di verificare il nome dell'autore, la data e l'orario delle firme, e autenticare il contenuto del messaggio, come le firme digitali, che offrono una sicurezza intrinseca superiore ed è pensata per risolvere il problema della manomissione e dell'imitazione (copiare intenzionalmente le caratteristiche di un'altra persona) nelle comunicazioni digitali.

In modo simile, il Transport Layer Security (TLS) è un tipo di protocollo di crittografia utilizzato per fornire sicurezza nella comunicazione su una rete. I siti web utilizzano il TLS per proteggere tutte le comunicazioni tra i loro server e i browser web. Una configurazione attenta del TLS fornisce anche ulteriori proprietà di privacy, garantendo che la futura divulgazione delle chiavi di crittografia non possa essere utilizzata per decifrare le comunicazioni TLS registrate in passato.

# L'angolo del geek

**Entità XML Esterne (XEE)**

Per coloro che sono appassionati di tecnologia, XML può essere oggetto di studio in quanto rappresenta un tipo di attacco nei confronti di un'applicazione web che analizza dati in formato XML. Questi dati XML possono fare riferimento a entità esterne, cercando di sfruttare una vulnerabilità nel parser. Un 'entità esterna' in questo contesto si riferisce a un'unità di archiviazione, come ad esempio un disco rigido. Un parser XML può essere ingannato nell'invio di dati a un'entità esterna non autorizzata, consentendo così a un attaccante di ottenere direttamente dati sensibili. [41]

I modi migliori per prevenire gli attacchi XEE sono quelli di far sì che le applicazioni web accettino tipi di dati meno complessi, come ad esempio JSON\*\*, oppure, almeno, di correggere i parser XML e disabilitare l'uso di entità esterne in un'applicazione XML.

\*XML o Extensible Markup Language è un linguaggio di markup progettato per essere leggibile sia dalle persone che dalle macchine. A causa della sua complessità e delle vulnerabilità di sicurezza, è attualmente in via di abbandono in molte applicazioni web.

\*\*JavaScript Object Notation (JSON) è un tipo di notazione semplice e leggibile dalle persone, spesso utilizzata per trasmettere dati su Internet. Anche se è stato originariamente creato per JavaScript, JSON è indipendente dal linguaggio e può essere interpretato da molti linguaggi di programmazione diversi.

**Per saperne di più:**

<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)

## Conclusioni

In questa unità, hai appreso varie modalità per comprendere come proteggere e progettare un sistema IT per soddisfare i suoi requisiti di sicurezza, bilanciando ciò con i suoi requisiti funzionali, come l'architettura della sicurezza funziona come un progetto di sicurezza unificato che affronta le necessità e i rischi potenziali in un determinato scenario o ambiente, quando e dove applicare controlli di sicurezza e altro ancora.

# Quiz

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1. **Cloud Security si riferisce a** | | | | | |
|  |  | |  |  |  |  |
|  | 1. Una vasta gamma di policies |  |  |  |  |  |
|  | 1. Technologie |  |  |  |  |  |
|  | 1. Deployment Controls |  |  |  |  |  |
|  | 1. **Tutte le indicate sopra**   **2. Cloud security protegge cosa?**   1. Data 2. Infrastruttura 3. Applicationi 4. **Tutte le indicate sopra** |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **3. I fornitori di servizi cloud si assicurano che ... tramite il cloud siano sicuri attraverso l'implementazione di procedure di test e accettazione per il codice di applicazioni esternalizzate o confezionate.** | | | | | |
|  |  | |  |  |  |
|  | 1. Piattaforma come servizio |  |  |  |  |
|  | 1. Infrastruttura come servizio |  |  |  |  |
|  | 1. Applicazioni disponibili come servizio |  |  |  |  |
|  | 1. Nessuna di queste   **4. Le principali preoccupazioni sulla sicurezza del cloud computing sono**   1. I dirigenti informatici faticano a fidarsi dei fornitori di servizi cloud relativamente nuovi e sconosciuti come parte delle loro estese imprese. 2. I professionisti dell'informatica si interrogano se i fornitori di servizi cloud abbiano i livelli di sicurezza dell'infrastruttura necessari per respingere gli attacchi informatici. 3. I fornitori hanno i mezzi necessari per gestire, misurare e segnalare le normative settoriali e possono essere ritenuti responsabili in caso di mancata conformità 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. **Tutte le indicate sopra** |  |  |  |  |

# Fonti

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]