



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica

DOCUMENTAZIONE PROGETTO DI PENETRATION TESTING  
AND ETHICAL HACKING

# **Un Framework per il Security Assessment di un'Infrastruttura Google Cloud Platform (GCP)**

**Francesco Giorgione** Matricola: 0522501741

Anno Accademico 2024-2025

---

## Indice

---

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Struttura del documento . . . . .	2
<b>2</b>	<b>Google Cloud Platform (GCP)</b>	<b>3</b>
2.1	Panoramica . . . . .	3
2.2	Organizzazione delle risorse . . . . .	6
2.2.1	Progetti . . . . .	7
2.2.2	Folder e organizzazioni . . . . .	7
2.3	Tipi di account . . . . .	8
2.3.1	Account utente (user account) . . . . .	8
2.3.2	Account di servizio (service account) . . . . .	8
2.4	Modalità di interazione con i servizi . . . . .	9
<b>3</b>	<b>Implementazione</b>	<b>11</b>
3.1	CIS GCP Foundation Benchmark . . . . .	12
3.2	Controlli implementati . . . . .	16
3.3	Ambiente GCP utilizzato . . . . .	26
<b>4</b>	<b>Istruzioni per l'utilizzo del framework</b>	<b>30</b>
4.1	Organizzazione della repository GitHub . . . . .	30
4.2	Utilizzo del framework . . . . .	30

<b>5 Conclusioni</b>	<b>33</b>
5.1 Sviluppi futuri . . . . .	33
<b>Bibliografia</b>	<b>34</b>

# CAPITOLO 1

---

## Introduzione

---

La crescente transizione di aziende, organizzazioni e pubbliche amministrazioni verso il cloud richiede di mettere in campo solide misure di sicurezza, necessarie per gestire in modo adeguato una vasta gamma di rischi, tra cui data breach, accessi non autorizzati, attacchi DoS. Tali rischi, tra l'altro, si inseriscono in un contesto che, per sua natura, è estremamente dinamico e di difficile gestione. Akinade et al. [1] forniscono una review di quelle che, allo stato dell'arte, rappresentano le *best practice* di cloud security. Tra le principali, viene citata la necessità di un'attività di *detection* e *prevention* in tempo reale, orchestrata dal SIEM, e di dotarsi di un Incident Response Plan preciso ed articolato. Ma è altrettanto importante focalizzarsi sulle c.d. vulnerabilità operative, i.e. derivanti da **errori di configurazione** e/o da un deploy improprio del sistema. Nell'ambito della c.d. *shared responsibility*, i cloud provider offrono una vasta gamma di strumenti di sicurezza, la cui configurazione, però, spetta tipicamente al client [2]. Pertanto, risulta cruciale analizzare dal punto di vista della sicurezza la configurazione dell'infrastruttura.

Questo progetto si propone di sviluppare un framework per il *Security Assessment* di un'infrastruttura Google Cloud Platform (GCP). Date le credenziali di un'infrastruttura, l'obiettivo è quello di analizzare, secondo una *checklist* dettagliata, le sue impostazioni di configurazione, al fine di individuare e segnalare criticità rispetto

alla postura di sicurezza.

## 1.1 Struttura del documento

Il documento è articolato in diversi capitoli, ognuno dei quali approfondisce aspetti specifici del lavoro svolto.

- Il Capitolo 2 fornisce una panoramica sulle caratteristiche di Google Cloud Platform, con particolare riferimento ai tipi di account e ruoli, nonché alla gerarchia delle risorse. Le informazioni sono estratte dalla documentazione ufficiale di GCP [3].
- Il Capitolo 3 descrive le finalità del progetto, le strategie implementative, l'architettura del framework creato e gli strumenti utilizzati.
- Il Capitolo 4 fornisce indicazioni in merito all'organizzazione della repository GitHub del progetto e all'utilizzo del framework.
- Il Capitolo 5 riassume le conclusioni emerse dal lavoro, proponendo possibili miglioramenti e sviluppi futuri.

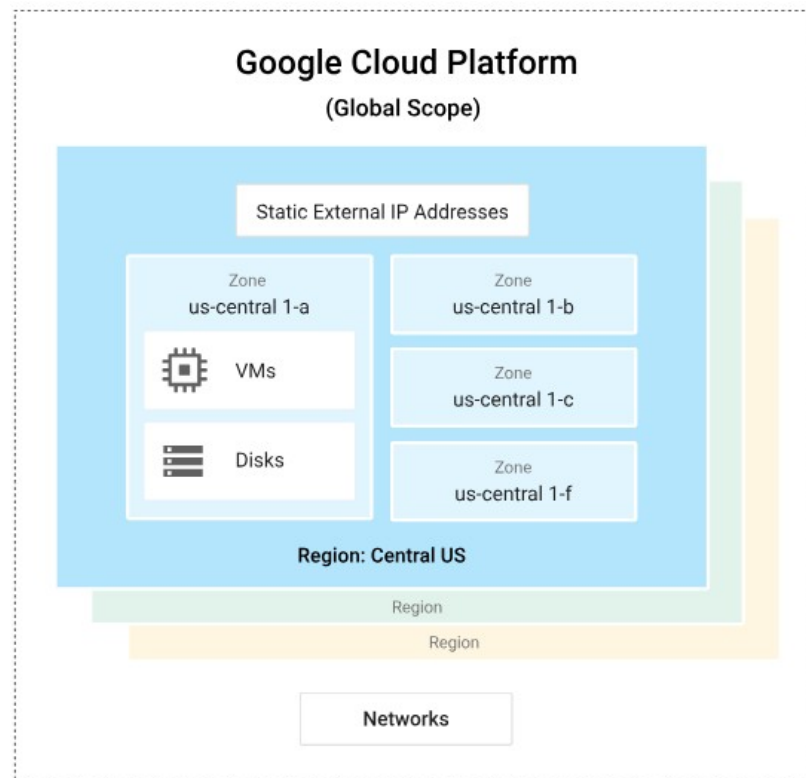
---

### Google Cloud Platform (GCP)

---

#### 2.1 Panoramica

Google Cloud Platform è la suite di servizi cloud offerta da Google per supportare aziende, sviluppatori e organizzazioni nella creazione, nella gestione e nella scalabilità di applicazioni e infrastrutture in modo sicuro, flessibile e ad alte prestazioni. Lanciata pubblicamente nel 2008, GCP si basa su un insieme di asset fisici come computer e dischi rigidi, nonché su risorse virtuali come le macchine virtuali (VM), ospitati nei data center di tutto il mondo. Ogni località dei data center si trova all'interno di una *regione*. Le regioni disponibili sono Asia, Australia, Europa, Africa, Medio Oriente, Nord America e Sud America. Ogni regione è composta da varie *zone*, isolate l'una dall'altra. Ogni zona è identificata da un nome che combina un identificatore di lettere con il nome della regione. Ad esempio, la zona *a* nella regione Asia orientale è denominata *asia-east1-a*. Il diagramma in Figura 2.1 mostra la relazione tra ambito globale, regioni, zone e alcune delle relative risorse. Le risorse sono fornite nell'ambito dell'erogazione di servizi. Le Tabelle 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7 forniscono una descrizione dei principali tipi di servizi, distinti per categorie.



**Figura 2.1:** Relazione tra ambito globale, regioni, zone e risorse [3]

Servizio	Descrizione
Compute Engine	Macchine virtuali altamente configurabili
App Engine	Piattaforma serverless per app web e API
Cloud Run	Esecuzione di container in modalità serverless
Google Kubernetes Engine (GKE)	Gestione di container su Kubernetes
Cloud Functions	Funzioni event-driven serverless

**Tabella 2.1:** Servizi Compute di GCP

Servizio	Descrizione
Vertex AI	Piattaforma unificata per addestramento e deployment di modelli ML
Vision AI / NLP / Speech APIs	API per riconoscimento immagini, testo e voce

**Tabella 2.2:** Servizi AI / ML di GCP

Servizio	Descrizione
Cloud Storage	Storage oggetti scalabile e durevole
Cloud SQL	Database relazionali gestiti (MySQL, PostgreSQL, SQL Server)
Firestore	Database NoSQL document-based in tempo reale
BigQuery	Data warehouse serverless per analisi su larga scala
Cloud Spanner	Database relazionale distribuito e scalabile

**Tabella 2.3:** Servizi di Storage e Database di GCP

Servizio	Descrizione
Virtual Private Cloud (VPC)	Rete privata isolata per le risorse cloud
Cloud Load Balancing	Bilanciamento di carico globale e regionale
Cloud NAT	Accesso internet in uscita da istanze private
Cloud VPN	Connessione sicura tra rete on-premise e VPC
Cloud Interconnect	Connessione fisica diretta a GCP
Cloud DNS	Servizio DNS altamente disponibile

**Tabella 2.4:** Servizi di Networking di GCP



Servizio	Descrizione
IAM	Gestione di permessi e ruoli per utenti e servizi
Cloud Identity	Identità e accesso per utenti e dispositivi
Cloud KMS	Servizio di gestione delle chiavi crittografiche
Security Command Center	Pannello di controllo della sicurezza cloud
Access Context Manager	Regole di accesso contestuale (es. IP, device)

**Tabella 2.5:** Servizi di Sicurezza e IAM di GCP

Servizio	Descrizione
Cloud Logging	Raccolta e analisi di log in tempo reale
Cloud Monitoring	Monitoraggio e allerta delle risorse
Cloud Build	Servizio CI/CD per costruzione container e codice
Artifact Registry	Registro per immagini Docker e pacchetti
Deployment Manager	Automazione del provisioning tramite template

**Tabella 2.6:** Servizi di Logging, Monitoring e DevOps di GCP

Servizio	Descrizione
Billing	Gestione e analisi dei costi
Resource Manager	Gestione di progetti, cartelle e organizzazioni
Policy Intelligence	Suggerimenti e audit per policy IAM
Service Usage	Abilitazione e monitoraggio dei servizi API

**Tabella 2.7:** Servizi di Governance e Gestione di GCP

## 2.2 Organizzazione delle risorse

Di seguito vengono descritti gli strumenti per l'organizzazione gerarchica delle risorse.

### 2.2.1 Progetti

In GCP, il progetto è l'unità organizzativa fondamentale con cui si gestiscono e isolano risorse, autorizzazioni, configurazioni. Per poter essere allocata, una risorsa deve necessariamente appartenere ad un progetto. Le risorse all'interno di un singolo progetto possono appartenere a regioni e zone diverse e possono comunicare tramite una rete interna, nel rispetto delle regole relative a regioni e zone. Un progetto non può accedere alle risorse di un altro progetto a meno che non utilizzi una *Virtual Private Cloud* (VPC). Un progetto funge da spazio dei nomi: ciascuna risorsa al suo interno deve avere un nome univoco. Ogni progetto GCP è caratterizzato da

- un **nome**, attribuito dall'utente creatore;
- un **ID**, attribuito dall'utente creatore o fornito da Google Cloud;
- un **numero**, fornito da Google Cloud.

L'ID del progetto è necessario per eseguire comandi e chiamate alle API. Ogni ID è univoco in GCP; anche se un progetto viene eliminato, il relativo ID non potrà mai più essere riutilizzato.

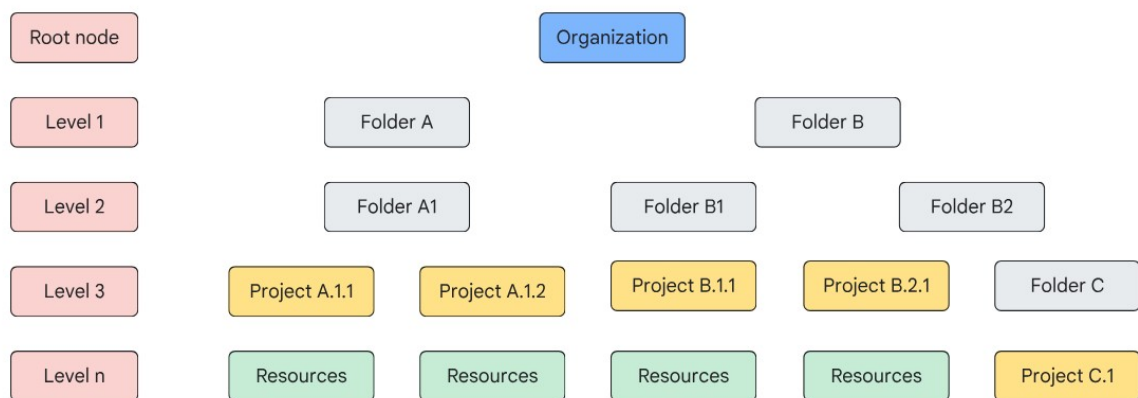
### 2.2.2 Folder e organizzazioni

Folder e organizzazioni sono entità gerarchiche particolarmente utili in contesti multi-progetto e multi-team.

L'**organizzazione** rappresenta l'entità aziendale principale (e.g. un'azienda, un'università) ed è il nodo radice della gerarchia GCP. Per creare un'organizzazione, è necessario disporre (o essere amministratore con accesso al DNS) di un dominio aziendale verificato da Google, e.g. *azienda.com*.

La **folder** è un livello organizzativo intermedio, opzionalmente usato per raggruppare progetti (o altre folder) all'interno di un'organizzazione. Una folder deve obbligatoriamente appartenere ad un'organizzazione: non può esistere come livello organizzativo autonomo. In Figura 2.2 è riportata una possibile organizzazione gerarchica delle risorse. Organizzazioni e folder sono livelli organizzativi **opzionali**. Infatti, è possibile

- creare un progetto al di fuori di un'organizzazione;
- creare un progetto direttamente all'interno di un'organizzazione, senza che esso appartenga ad alcuna folder.



**Figura 2.2:** Esempio di gerarchia delle risorse

Uno dei principali vantaggi dell'utilizzo di folder e organizzazioni è la **gestione centralizzata della sicurezza**: è possibile assegnare ruoli IAM (*Identity and Access Management*) a livello di organizzazione, folder o progetto; le policy vengono ereditate lungo la gerarchia, dall'alto verso il basso.

## 2.3 Tipi di account

In GCP esistono principalmente due tipi di account: quelli che rappresentano persone (**utenti**) e quelli che rappresentano **applicazioni** o **servizi**.

### 2.3.1 Account utente (user account)

Sono account gestiti da una persona reale. Possono avere dominio aziendale (e.g. nome@azienda.com) o essere account Gmail personali (e.g. nome@gmail.com).

### 2.3.2 Account di servizio (service account)

Sono account non umani, gestiti e utilizzati da applicazioni, VM, container, script o servizi GCP per eseguire azioni in automatico. Anche gli account di servizio sono identificati da un indirizzo email univoco, della forma `<my-service-account>@<my-project>.iam.gserviceaccount.com`. Quando un'applicazione si autentica come account di servizio, ha accesso a tutte le risorse a cui l'account di servizio è autorizzato ad accedere. Il modo più comune per consentire a un'applicazione di autenticarsi come account di servizio è collegare un account di servizio alla risorsa che esegue l'applicazione. Ad esempio, è possibile collegare un account di servizio

a un'istanza Compute Engine in modo che le applicazioni in esecuzione su quell'istanza possano autenticarsi come account di servizio. Diversamente dagli account utente, tali account non hanno una password, ma utilizzano coppie di chiavi RSA per l'autenticazione. Ogni service account può possedere le due seguenti categorie di chiavi di servizio.

- **Chiavi GCP-managed:** sono create e gestite automaticamente da Google per essere utilizzate da servizi interni, e.g. Compute Engine, App Engine; non sono visibili né scaricabili; vengono ruotate automaticamente ogni 7 giorni e sono sicure di default.
- **Chiavi user-managed:** sono create manualmente dall'utente tramite console o CLI; possono essere scaricate; hanno una scadenza di default di 10 anni e devono essere gestite attivamente da parte dell'utente (e.g. rotazione, revoca, protezione).

Esistono i seguenti tipi di account di servizio.

- **Gestiti dall'utente (*user-managed*):** l'utente è responsabile della loro gestione. Si classificano ulteriormente in
  - **account di default:** sono creati automaticamente da Google quando si attivano determinati servizi;
  - **account user-created:** sono creati dall'utente per propri scopi.
- **Agenti di servizio (*Google-managed*):** sono creati e gestiti da Google Cloud; non è possibile associarvi chiavi di servizio user-managed.

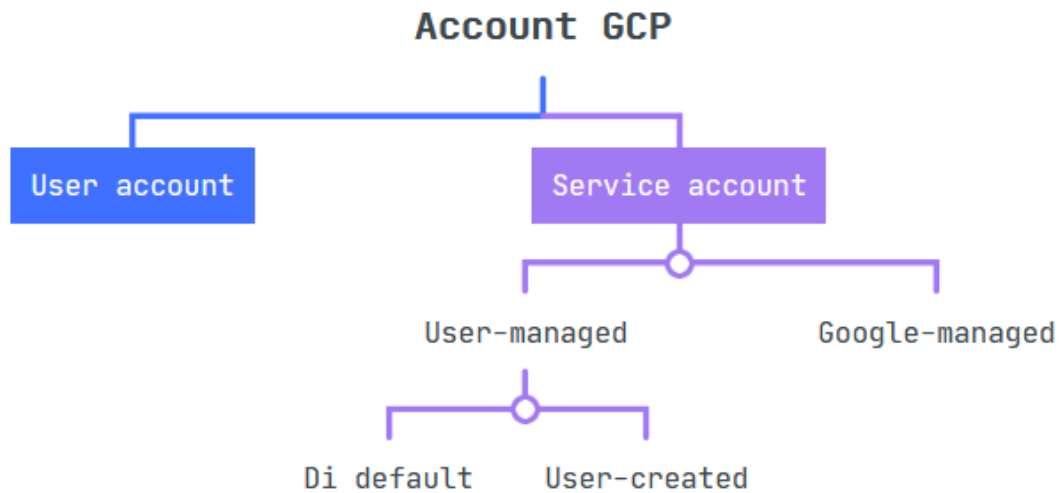
Lo schema in Figura 2.3 sintetizza quanto appena spiegato.

## 2.4 Modalità di interazione con i servizi

Google Cloud offre tre modalità di base per interagire con i servizi e le risorse.

### Google Cloud Console

È una GUI web-based molto intuitiva, utile per gestire, configurare e monitorare tutte le risorse e i servizi. Fornisce una dashboard centralizzata da cui puoi controllare l'intera infrastruttura cloud, senza usare comandi da terminale.



**Figura 2.3:** Tipi di account GCP

## gcloud

È un'interfaccia a linea di comando, che consente di eseguire tutte le operazioni eseguibili tramite console, ma in modo più rapido ed automatizzabile. I comandi gcloud possono essere eseguiti nei seguenti modi:

- tramite **Google Cloud CLI** (da installare), aprendo in locale una finestra del terminale;
- tramite **Google Cloud Shell**, i.e. una shell ottenibile direttamente dalla console, senza necessità di installazioni.

Google Cloud Shell fornisce, tra le altre cose, un'istanza temporanea di Virtual Machine (VM), un editor integrato, funzionalità di anteprima web e un'autorizzazione integrata per l'accesso a progetti e risorse.

## API

Google fornisce librerie client per vari linguaggi - e.g. JavaScript, Python, Java, C# - per interrogare le API di GCP. Queste ultime sono ottimizzate per i linguaggi supportati e forniscono anche funzionalità amministrative.

## CAPITOLO 3

---

### Implementazione

---

Come già spiegato, l'obiettivo del presente progetto è quello di sviluppare un framework per il *Security Assessment* di un'infrastruttura GCP, con particolare riguardo alle impostazioni di configurazione. Tuttavia, progettare un meccanismo di *audit* basato su checklist è tutt'altro che banale. Le infrastrutture cloud moderne sono composte da risorse eterogenee, ognuna con proprie configurazioni di sicurezza, permessi e dipendenze. A ciò si aggiungono peculiarità tipiche del modello cloud, come la natura dinamica delle risorse, la condivisione della responsabilità tra provider e cliente, l'uso diffuso di identità e ruoli granulari per il controllo degli accessi. Questi fattori rendono complesso ottenere una visione completa e coerente del livello di sicurezza dell'intera infrastruttura. In un contesto così complicato, risulta essenziale procedere in modo sistematico, tenendo conto di standard e benchmark esistenti allo stato dell'arte. In particolare, **CIS (Center for Internet Security)** pubblica i c.d. **CIS Benchmarks** [4], i.e. checklist per il security assessment della configurazione di infrastrutture cloud di specifici provider, tra cui GCP, Azure, AWS, Alibaba Cloud. Per il presente progetto, si è ritenuto di usare questo strumento, in quanto, per ciascun controllo di sicurezza, esso fornisce una dettagliata descrizione di come implementarlo e informazioni su strategie di mitigazione. In produzione, è fondamentale integrare i controlli tecnici del CIS Benchmarks tramite strumenti eterogenei, tra cui, a titolo esemplificativo:

- **CSA CCM (Cloud Controls Matrix)**: matrice per il security assessment di ambienti e provider cloud; include controlli organizzati per domini, e.g. governance, IAM,

network security;

- **Linee guida del NIST**, come la SP 800-144, avente ad oggetto *Guidelines on Security and Privacy in Public Cloud Computing*;
- **ISO/IEC 27017**, avente ad oggetto *Controlli di sicurezza per i Servizi Cloud*;
- **ISO/IEC 27018**, i.e. *Codice di condotta per la protezione delle PII (Personally Identifiable Information) nei servizi di public cloud per i cloud provider*;
- **OWASP Cloud-Native Application Security Top 10**.

Inoltre, è necessario considerare il contesto normativo, con particolare attenzione alle leggi sulla privacy.

### 3.1 CIS GCP Foundation Benchmark

È un documento tecnico che elenca configurazioni di sicurezza consigliate per le risorse GCP, con l'obiettivo di ridurre la superficie d'attacco e prevenire configurazioni errate. Per il presente progetto si utilizza la **versione 4.0.0** (pubblicata il 5 Febbraio 2025), che risulta essere la più recente allo stato dell'arte. Le raccomandazioni fornite sono suddivise in 8 sezioni, per ciascuna delle quali, in Tabella 3.1, è riportato il numero di raccomandazioni contenute.

Categoria di raccomandazioni	Numero di raccomandazioni
Identity and Access Management (IAM)	17
Logging and Monitoring	16
Networking	10
Virtual Machines	12
Storage	2
Cloud SQL Database Services	22
BigQuery	4
Dataproc	1

**Tabella 3.1:** Categorie e numero di raccomandazioni nel CIS GCP Benchmark

Per ciascuna raccomandazione, sono riportati i seguenti elementi:

- a) una descrizione;

- b) il razionale;
- c) l’impatto, e.g. maggiore complessità organizzativa;
- d) **la lista di comandi Google Cloud CLI per verificare che la raccomandazione sia rispettata;**
- e) una strategia di *remediation*, qualora la raccomandazione non sia rispettata;
- f) le misure di prevenzione;
- g) il comportamento di default di GCP rispetto all’oggetto della raccomandazione;
- h) una lista di riferimenti a documentazioni varie;
- i) i CIS Controls, i.e. un insieme di best practice rispetto all’oggetto della raccomandazione.

Di particolare interesse è la voce (d), in quanto la lista di comandi fornita può essere utilizzata per costruire uno **script bash** che automatizzi la verifica. La parte implementativa del presente progetto consiste proprio in questo: individuato un sottoinsieme di raccomandazioni, per ciascuna di esse viene fornito uno script bash che esegue la verifica sull’infrastruttura ricevuta in input e segnala l’eventuale inottemperanza alla raccomandazione stessa. Ciascuno script restituisce, come valore di ritorno

- **0**, se l’esecuzione termina senza errori e nessun elemento di non conformità è stato individuato;
- **1**, se lo script termina in una condizione di errore indipendente dall’eventuale presenza di elementi di non conformità;
- **2**, se lo script, pur terminando senza errori, individua elementi di non conformità, che vengono esplicitamente segnalati;
- **3**, se lo script non riesce ad invocare le API di GCP a causa delle limitazioni di rate nella versione gratuita.

Gli script creati vengono messi a sistema dallo script *create-all.sh*, che riceve in input l’ID di un progetto GCP e provvede ad invocarli tutti, producendo in output i seguenti artefatti:

- un **file di log**, ottenuto dalla concatenazione degli output prodotti da ciascuno script invocato;
- un **report in formato HTML**, a sua volta contenente
  - una **prima tabella** che indica l’esito di ciascuno dei controlli condotti;



```

=====
Running script: 4-6.sh for project: pteh-04
=====

Updated property [core/project].
Checking instances for IP forwarding...

All instances have IP forwarding disabled.

Script 4-6.sh exited with code: 0

-----
End of script: 4-6.sh
-----

=====
Running script: 4-9.sh for project: pteh-04
=====

Updated property [core/project].
Checking for non-GKE instances with public IPs in project 'pteh-04'...
NON-COMPLIANT: The following non-GKE instances are set to have public IP addresses
vm-free-1
vm-free-2

Script 4-9.sh exited with code: 2

```

**Figura 3.1:** Estratto del file di log: indicazione dell’esito dei controlli 4.6 e 4.9 del CIS Benchmarks per GCP

- una **seconda tabella**, che riporta, per ciascuna categoria, il numero totale di controlli effettuati con successo (exit code 0 oppure 2), il numero di controlli non superati, il tasso di non conformità (dato dal rapporto tra i due valori precedenti) e un grafico a torta a supporto.

In Figura 3.1 è riportato l’output (parziale) di un file di log prodotto dal framework. Le Figure 3.2 e 3.3, invece, rappresentano un esempio del contenuto (parziale) del report HTML. Come già accennato, l’utilizzo delle API, nella versione gratuita, è limitato in termini di numero di richieste eseguibili in un intervallo temporale predefinito, peraltro secondo meccanismi eterogenei a seconda della specifica tipologia di risorsa. Ecco perché si è resa necessaria l’implementazione, nei vari script, di un meccanismo di timeout (gestito tramite

## Script Execution Report

Script	Status
1-1.sh	Warning: elements of non-compliance detected!
1-4.sh	Warning: elements of non-compliance detected!
1-5.sh	OK! All settings are compliant!
1-6.sh	Warning: elements of non-compliance detected!
2-1.sh	Warning: elements of non-compliance detected!
2-3.sh	Execution failed due to free GCP API rate limitation. Try again!
3-1.sh	Execution failed due to free GCP API rate limitation. Try again!
4-6.sh	OK! All settings are compliant!
4-9.sh	Warning: elements of non-compliance detected!
6-1-3.sh	Execution failed due to free GCP API rate limitation. Try again!

Figura 3.2: Estratto del report HTML: indicazione dell'esito di ciascun controllo

## Category Summary




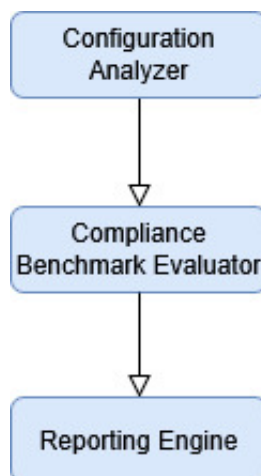
Category	Total Checks	Non-Compliant	Non-Compliance Rate (%)	Graph
4	2	1	50.00%	<p>Category 4</p>  <p>Compliant Non-Compliant</p>
2	1	1	100.00%	<p>Category 2</p>  <p>Compliant Non-Compliant</p>
1	4	3	75.00%	<p>Category 1</p>  <p>Compliant Non-Compliant</p>

Figura 3.3: Estratto del report HTML: visualizzazione di informazioni statistiche per categoria di controlli



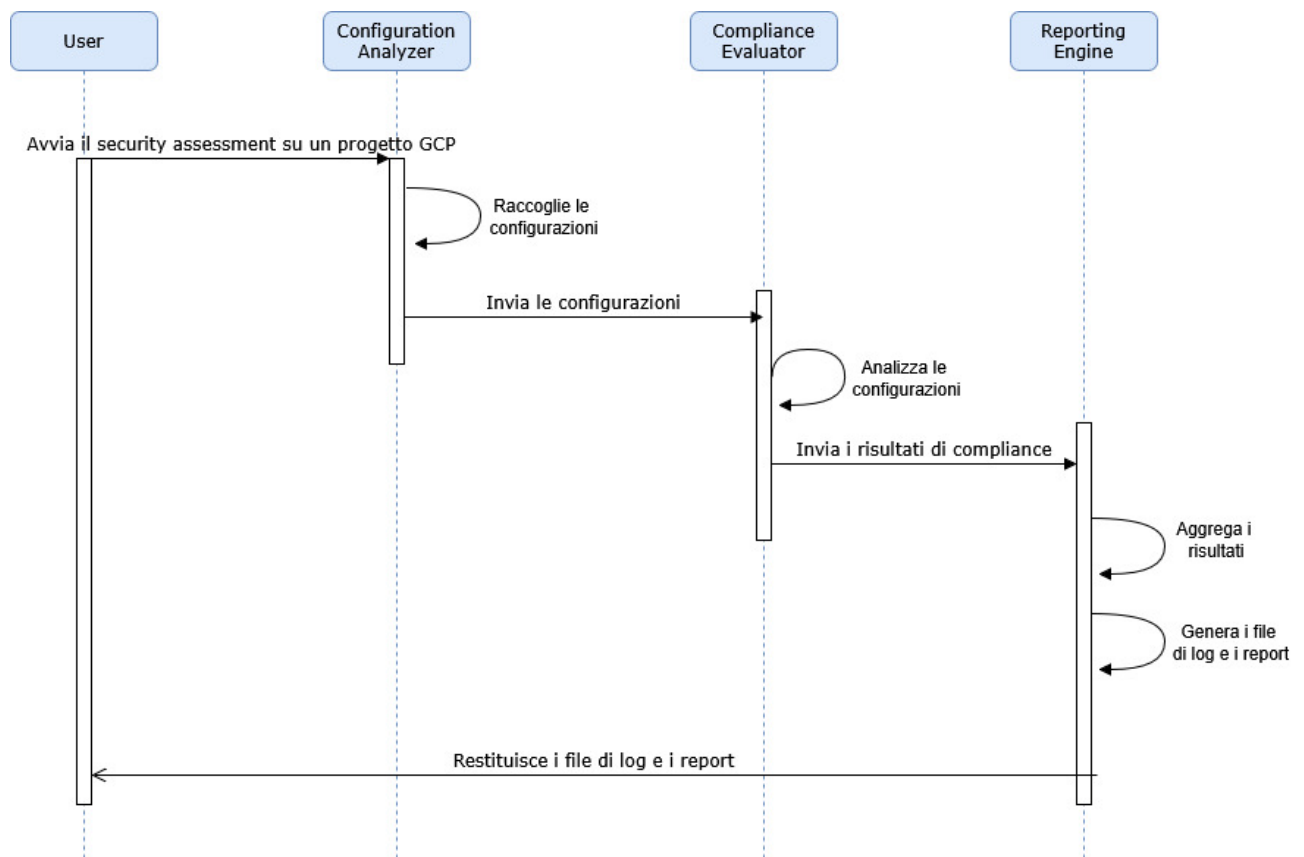
**Figura 3.4:** Component diagram: architettura ad alto livello del framework

l'exit code 3). In caso contrario, il fallimento di un'invocazione determinerebbe l'interruzione dell'intero flusso di verifica. Si noti che l'implementazione è **altamente modulare**, in quanto ciascun controllo è eseguibile in modo indipendente, mediante l'invocazione del relativo script. Ciò è utile, a maggior ragione, al verificarsi di timeout: è possibile ritentare l'esecuzione soltanto degli script per i quali si è precedentemente ottenuto l'exit code 3. Il component diagram in Figura 3.4 descrive l'architettura ad alto livello del framework creato: i singoli script di controllo mappano le funzionalità delle componenti *Configuration Analyzer* e *Compliance Benchmark Evaluator*; lo script *check-all.sh*, che si occupa di combinare i risultati per generare i log e i report, assolve al ruolo di *Reporting Engine*. Si osservi che, grazie al fatto che i comandi Google Cloud CLI consentono di verificare direttamente il rispetto delle varie raccomandazioni, non è necessaria una componente *Collector* che si occupi preliminarmente di collezionare le risorse. L'interazione tra le diverse componenti e l'utente è rappresentata dal sequence diagram in Figura 3.5.

La sezione successiva descrive in modo più dettagliato i controlli implementati.

## 3.2 Controlli implementati

La presente sezione descrive in maniera più dettagliata i controlli implementati, di cui viene innanzitutto fornita una panoramica in Tabella 3.2. Le numerazioni di seguito utilizzate per le varie raccomandazioni sono quelle riportate nel CIS GCP Benchmark.



**Figura 3.5:** Sequence diagram: interazione nel tempo tra le diverse componenti

Categoria di raccomandazioni	Controlli implementati
Identity and Access Management (IAM)	<p><b>1.1</b> Ensure that Corporate Login Credentials are Used</p> <p><b>1.4</b> Ensure That There Are Only GCP-Managed Service Account Keys for Each Service Account</p> <p><b>1.5</b> Ensure That Service Account Has No Admin Privileges</p> <p><b>1.6</b> Ensure That IAM Users Are Not Assigned the Service Account User or Service Account Token Creator Roles at Project Level</p>
Logging and Monitoring	<p><b>2.1</b> Ensure That Cloud Audit Logging Is Configured Properly</p> <p><b>2.3</b> Ensure That Retention Policies on Cloud Storage Buckets Used for Exporting Logs Are Configured Using Bucket Lock</p>
Networking	<b>3.1</b> Ensure That the Default Network Does Not Exist in a Project
Virtual Machines	<p><b>4.6</b> Ensure That IP Forwarding Is Not Enabled on Instances</p> <p><b>4.9</b> Ensure That Compute Instances Do Not Have Public IP Addresses</p>
Storage	-
Cloud SQL Database Services	<b>6.1.3</b> Ensure That the 'Local_infile' Database Flag for a Cloud SQL MySQL Instance Is Set to 'Off'
BigQuery	-
Dataproc	-

**Tabella 3.2:** Indicazione delle raccomandazioni verificate per ciascuna categoria

## 1.1 Ensure that Corporate Login Credentials are Used

Si raccomanda di accedere all'infrastruttura tramite account aziendali, piuttosto che personali (e.g. con dominio @gmail.com). L'utilizzo di account aziendali, infatti, semplifica il monitoraggio, l'auditing e il controllo degli accessi. Al contrario, gli account consumer non sono gestibili centralmente dall'organizzazione e non possono essere facilmente monitorati o revocati. Il comando seguente restituisce la IAM policy del progetto specificato, da cui si evince la lista degli account che hanno il permesso di accedere allo stesso.

```
gcloud projects get-iam-policy PROJECT_ID
```

In alternativa, il comando seguente restituisce la lista degli account che hanno il permesso di accedere alla *folder* specificata.

```
gcloud resource-manager folders get-iam-policy FOLDER_ID
```

Infine, il comando seguente restituisce la lista degli account che hanno il permesso di accedere all'*organizzazione* specificata.

```
gcloud organizations get-iam-policy ORGANIZATION_ID
```

È stato quindi creato uno script, basato sui precedenti comandi, che, presi in input il nome del progetto (oppure della folder/organizzazione) e il dominio autorizzato, verifica se utenti esterni al dominio specificato hanno accesso al progetto specificato. In caso affermativo, viene stampato un messaggio di errore.

## 1.4 Ensure That There Are Only GCP-Managed Service Account Keys for Each Service Account

È raccomandato che i service account non abbiano chiavi di servizio user-managed, ma soltanto GCP-managed; da 2.3.2, sappiamo che la gestione di queste ultime è realizzata in modo sicuro da GCP. Al contrario, le chiavi user-managed potrebbero essere gestite in modo poco sicuro. Soltanto a titolo esemplificativo, potrebbero finire nel codice, essere memorizzate su macchine non sicure, non essere mai ruotate, rimanere attive anche dopo anni. Il comando seguente restituisce la lista di tutti gli account di servizio.

```
gcloud iam service-accounts list
```

Il seguente comando, invece, fornisce la lista delle chiavi di servizio user-managed associate all'account specificato. Secondo quanto raccomandato, nessuna chiave dovrebbe essere mostrata.

```
gcloud iam service-accounts keys list --iam-account=<Service
Account> --managed-by=user
```

Utilizzando i due precedenti comandi, è stato creato uno script che, preso in input il nome del progetto, verifica l'esistenza di service account user-managed aventi chiavi di servizio user-managed. In caso affermativo, viene stampato un messaggio di errore. Tale script sfrutta la peculiarità dei soli service account user-managed di terminare sempre per @<PROJECT\_ID>.iam.gserviceaccount.com.

## 1.5 Ensure That Service Account Has No Admin Privileges

È raccomandato che i service account non abbiano privilegi elevati (da amministratore). Infatti, essendo i service account gli strumenti tramite cui le applicazioni accedono ad API e servizi, assegnare il ruolo di amministratore ad un account di servizio consente a tutte le applicazioni che lo usano di eseguire operazioni con privilegi elevati. La raccomandazione va applicata ai soli account di servizio user-managed e user-created, i cui indirizzi sono della forma <SERVICE\_ACCOUNT\_NAME>@<PROJECT\_ID>.iam.gserviceaccount.com. Al contrario, molti account di servizio Google-managed necessitano di privilegi elevati per funzionare correttamente. Come specificato dalla raccomandazione, i ruoli con privilegi elevati sono quelli il cui nome soddisfa una delle condizioni seguenti: (a) contiene *Admin* oppure *admin*; (b) è *roles/editor*; (c) è *roles/owner*. Il seguente comando restituisce la lista, in formato JSON, degli account che hanno il permesso di accedere al progetto specificato.

```
gcloud projects get-iam-policy PROJECT_ID --format json
```

Un esempio del risultato prodotto dal precedente comando è il seguente:

```
{
  "bindings": [
    {
      "members": [
        "serviceAccount:our-project-123@appspot.gserviceaccount.com",
      ],
    }
  ]
}
```

```

    "role": "roles/appengine.appAdmin"
  },
  {
    "members": [
      "user:email1@gmail.com"
    ],
    "role": "roles/owner"
  },
  {
    "members": [
      "serviceAccount:our-project-123@appspot.gserviceaccount.com",
      "serviceAccount:123456789012-compute@developer.
        gserviceaccount.com"
    ],
    "role": "roles/editor"
  }
],
"etag": "BwUjMhCsNvY=",
"version": 1
}

```

È stato quindi creato uno script, basato sul comando precedente, che, preso in input il nome del progetto, verifica l'esistenza di service account user-managed e user-created aventi privilegi elevati. In caso affermativo, viene stampato un messaggio di errore.

## 1.6 Ensure That IAM Users Are Not Assigned the Service Account User or Service Account Token Creator Roles at Project Level

Qualora si voglia assegnare ad un utente il ruolo di Service Account User (*iam.serviceAccountUser*) oppure il ruolo di Service Token Creator (*iam.serviceAccountTokenCreator*), si raccomanda di farlo a livello di specifico service account, piuttosto che a livello di progetto. Di seguito viene fornita una breve descrizione dei suddetti ruoli:

- **iam.serviceAccountUser:** consente all'utente cui è assegnato di collegare un account di servizio a una risorsa, di fatto permettendo all'account di servizio di eseguire



operazioni su quella risorsa;

- **iam.serviceAccountTokenCreator**: consente all'utente cui è attribuito di creare credenziali di breve durata per un account di servizio.

Attribuire tali ruoli a livello di progetto consente all'utente di eseguire le precedenti operazioni privilegiate su un qualunque account di servizio del progetto, inclusi quelli eventualmente creati dopo l'attribuzione del ruolo. Pertanto, è raccomandato agire in modo più granulare, assegnando ruoli di questo tipo a livello di singolo service account. Più in generale, la sicurezza dei servizi è determinata dai ruoli IAM associati ai relativi account di servizio, per cui è fondamentale prestare la giusta attenzione alla configurazione di questi ultimi.

Il comando seguente ottiene, in formato JSON, la policy IAM **a livello di progetto** ed estrae il valore del campo *role* di ogni *member*. Per comprendere la struttura della policy IAM ricavata, si può far riferimento all'esempio JSON riportato in 3.2. Infine, tramite l'applicazione di un filtro, vengono stampati soltanto i valori *role* che contengono la stringa *roles/iam.serviceAccountUser*. Pertanto, l'output di tale comando è vuoto se e solo se a nessun utente è stato attribuito il ruolo di Service Account User a livello di progetto.

```
gcloud projects get-iam-policy PROJECT_ID --format json | jq
'.bindings[].role' | grep "roles/iam.serviceAccountUser"
```

Analogamente, l'output del seguente comando è vuoto se e solo se a nessun utente è stato attribuito il ruolo di Service Account Token Creator a livello di progetto.

```
gcloud projects get-iam-policy PROJECT_ID --format json | jq
'.bindings[].role' | grep "roles/iam.serviceAccountTokenCreator"
```

A partire dai precedenti comandi, è stato creato uno script che, preso in input il nome del progetto, verifica l'esistenza di ruoli di tipo Service Account User e Service Account Token Creator a livello di progetto.

## 2.1 Ensure That Cloud Audit Logging Is Configured Properly

Si raccomanda di configurare il servizio Cloud Audit Logging per assicurare che vengano tracciate tutte le attività amministrative, nonché tutte le operazioni di lettura, scrittura ed accesso ai dati utente. La configurazione opportuna richiede che

- *logtype* sia settato a ADMIN\_READ, DATA\_READ e DATA\_WRITE;
- *audit config* sia abilitato per tutti i servizi che supportano le feature di logging;

- i log siano catturati per tutti gli utenti, i.e. non siano dichiarati utenti *exempted*.

Il seguente comando restituisce la IAM policy del progetto specificato.

```
gcloud projects get-iam-policy PROJECT_ID
```

Nell'apposita sezione dell'output, sono riportate le opzioni di configurazione relative al logging. Secondo quanto spiegato in precedenza, la configurazione opportuna è della forma seguente.

```
auditConfigs:
- auditLogConfigs:
- logType: ADMIN_READ
- logType: DATA_WRITE
- logType: DATA_READ
  service: allServices
```

Si noti che il parametro *exemptedMembers* non è impostato, per cui il logging è abilitato per tutti gli utenti, senza alcuna eccezione. A partire dal precedente comando, dunque, è stato creato uno script che verifica che i campi della sezione *auditConfigs* siano impostati come sopra, e che l'argomento *exemptedMembers* non sia impostato. In caso contrario, viene stampato un messaggio di errore.

## 2.3 Ensure That Retention Policies on Cloud Storage Buckets Used for Exporting Logs Are Configured Using Bucket Lock

In GCP, è possibile esportare i log da Cloud Logging verso Cloud Storage, utilizzando i c.d. sink di logging. Essi specificano (a) un filtro, per selezionare i log da esportare; (b) una destinazione, e.g. un bucket Cloud Storage. Le motivazioni a favore dell'esportazione sono molteplici. La principale è che Cloud Logging ha una retention predefinita limitata: viceversa, l'esportazione in un bucket consente la memorizzazione dei log per periodi molto più lunghi. In alcuni settori, la memorizzazione a lungo termine dei log è, peraltro, obbligatoria per legge. Inoltre, i log esportati possono essere elaborati con strumenti di analisi avanzata e data mining. È fondamentale proteggere i log esportati, evitandone cancellazioni e/o alterazioni, accidentali o volontarie. A tale scopo, Cloud Storage offre la funzionalità di *Bucket Lock*, che consente di **bloccare una politica di retention**, rendendola immutabile. I log esportati nel

bucket non possono essere eliminati prima della scadenza della retention, neanche se si è in possesso dei massimi privilegi.

I seguenti comandi consentono, rispettivamente, di ottenere la lista dei sink esistenti e verificare che il sink specificato abbia una policy di retention attiva.

```
gcloud logging sinks list --project=PROJECT_ID
gsutil retention get gs://BUCKET_NAME
```

È stato quindi creato uno script, basato sui precedenti comandi, che, preso in input il nome del progetto, verifica l'esistenza di bucket associati ai sink per i quali non sia stata prevista alcuna policy di retention o per i quali sia stata prevista una policy di retention priva di lock. In caso affermativo, viene stampato un messaggio di errore.

### 3.1 Ensure That the Default Network Does Not Exist in a Project

A meno che non sia esplicitamente disabilitata, ogni progetto possiede una rete VPC di default, detta *default network*. Essa fornisce impostazioni di configurazione predefinite, tra cui le seguenti regole firewall IPv4, notevolmente insicure:

- **default-allow-internal**: consente connessioni in ingresso per tutti i protocolli e tutte le porte tra le istanze interne alla VPC;
- **default-allow-ssh**: consente connessioni in ingresso sulla porta 22 TCP (default per il servizio SSH) da ogni sorgente (anche esterna) ad ogni destinazione interna alla VPC;
- **default-allow-rdp**: consente connessioni in ingresso sulla porta 3389 TCP (default per il servizio RDP, i.e. *Remote Desktop Protocol*) da ogni sorgente (anche esterna) ad ogni destinazione interna alla VPC;
- **default-allow-icmp**: consente il traffico ICMP in ingresso da ogni sorgente (anche esterna) ad ogni destinazione interna alla VPC.

Tra l'altro, di default tali regole non generano log. Pertanto, il presente controllo raccomanda di eliminare la *default network* e di crearne una nuova.

Il comando seguente restituisce la lista delle reti disponibili per il progetto.

```
gcloud compute networks list
```

A partire dal precedente comando, è stato creato uno script che, preso in input il nome del progetto, verifica che la *default network* non compaia nella lista delle reti disponibili. In caso contrario, viene stampato un messaggio di errore.

## 4.6 Ensure That IP Forwarding Is Not Enabled on Instances

Di default, le istanze Computer Engine (i.e. le VM) non possono eseguire il forward di un pacchetto, a meno che l'IP sorgente del pacchetto non coincida con l'IP della VM. Analogamente, GCP esegue il delivery di un pacchetto a una VM solo se l'IP di destinazione del pacchetto coincide con l'IP della VM. In altre parole, le VM non possono fare routing. L'instradamento a livello di VM, che pure potrebbe risultare utile per efficientare il routing, comporta inevitabilmente rischi di data loss e information disclosure. Pertanto, il presente controllo raccomanda di disabilitarlo, impostando a *false* il campo *canIpForward* sulle istanze Computer Engine.

Il seguente comando mostra, per ogni istanza Computer Engine, il valore del campo *canIpForward*.

```
gcloud compute instances list --format='table(name,canIpForward)'
```

A partire dal precedente comando, è stato costruito uno script che, preso in input il nome del progetto, verifica l'esistenza di VM per le quali è abilitato l'IP forwarding. In tal caso, viene stampato un messaggio di errore.

## 4.9 Ensure That Compute Instances Do Not Have Public IP Addresses

Al fine di ridurre la superficie di attacco, è raccomandato che le VM non abbiano un indirizzo IP pubblico. Piuttosto, se necessitano di esporre servizi, dovrebbero essere configurate dietro un load balancer, al fine di ridurre l'esposizione. La raccomandazione non vale per le istanze GKE, i.e. create e gestite automaticamente da Google Kubernetes Engine (GKE) per eseguire i nodi di un cluster Kubernetes. Tali istanze soddisfano le due proprietà seguenti: (a) il loro nome comincia con *gke-*; (b) hanno un'etichetta *goog-gke-node*.

Il seguente comando restituisce una lista, in formato JSON, delle istanze Computer Engine e delle relative proprietà.

```
gcloud compute instances list --format=json
```

Un'istanza non è configurata per essere pubblicamente esposta se e solo se, all'interno dell'oggetto JSON restituito, il relativo elemento **non** possiede il campo `accessConfigs.networkInterfaces`. Pertanto, a partire dal precedente comando, è stato costruito uno script che verifica l'esistenza di tale campo, prestando attenzione all'eccezione per le istanze

create da GKE. Qualora individui istanze configurate per avere un indirizzo IP pubblico, lo script stampa un messaggio di errore.

### 6.1.3 Ensure That the 'Local\_infile' Database Flag for a Cloud SQL MySQL Instance Is Set to 'Off'

In MySQL, il flag *local\_infile* abilita o disabilita l'uso del comando seguente, che consente di caricare direttamente, in una tabella del database, i dati di un file ubicato sul computer del client.

```
LOAD DATA LOCAL INFILE 'percorso/del/file.csv' INTO TABLE
  nome_tabella;
```

In passato, MySQL è stato vulnerabile a exploit che usavano il comando `LOAD DATA LOCAL INFILE` per rubare file sensibili. Pertanto, il presente controllo raccomanda di impostare a *off* il flag *local\_infile*, a meno di specifiche motivazioni.

Il comando seguente restituisce la lista di tutte le istanze di database CloudSQL.

```
gcloud sql instances list
```

Di seguito, invece, è riportato il comando che, data una specifica istanza di DB Cloud SQL, stampa il valore del flag *local\_infile*.

```
gcloud sql instances describe <INSTANCE_NAME> --format=json | jq '.
  settings.databaseFlags[] | select(.name=="local_infile")|.value'
```

A partire dai due precedenti comandi, è stato costruito uno script che, preso in input il nome del progetto, verifica che, per ciascuna istanza di DB Cloud SQL, il flag *local\_infile* non sia esplicitamente impostato a *on*. Qualora il flag non sia dichiarato, il comando `LOAD DATA LOCAL INFILE` è disabilitato di default.

## 3.3 Ambiente GCP utilizzato

Al fine di testare il framework implementato, sono stati creati **due diversi progetti** GCP, aventi le stesse risorse ma configurazioni diverse. I loro ID, univoci in GCP, sono *pteh-02* e *pteh-04*. Entrambi i progetti sono esterni a qualsiasi organizzazione e folder: non si è potuto fare altrimenti, dato che, come già spiegato in 2.2.2, la creazione di un'organizzazione richiede

Tipo di risorsa	Servizio	# Istanze	Descrizione
Instance/Virtual Machine (VM)	Compute Engine	2	Macchine E2-micro, i.e. a basse prestazioni, che rientrano nel free tier
Bucket	Cloud Storage	2	Aree di archiviazione file, gratuite fino a 5 GB/mese
Firestore	Firestore	1	DB NoSQL serverless e ad alte prestazioni
Database SQL	Cloud SQL	1	DB MySQL v 5.7

**Tabella 3.3:** Categorie e numero di raccomandazioni nel CIS GCP Benchmark

Raccomandazione del CIS GCP Benchmark	Rispettata nel Progetto 1	Rispettata nel Progetto 2	Script per l'introduzione degli elementi di non conformità
1.1	NO	NO	<i>default</i>
1.4	NO	NO	create-servaccounts.sh
1.5	NO	SI	create-servaccounts.sh
1.6	NO	NO	create-user.sh
2.1	NO	NO	add-audit-config.sh
2.3	NO	SI	create-sinks.sh
3.1	NO	NO	<i>default</i>
4.6	NO	SI	create-resources.sh
4.9	NO	NO	<i>default</i>
6.1.3	NO	SI	create-resources.sh

**Tabella 3.4:** Indicazione delle raccomandazioni rispettate nei due progetti e degli script utilizzati per l'introduzione degli elementi di non conformità

## Script Execution Report

Script	Status
1-1.sh	Warning: elements of non-compliance detected!
1-4.sh	Warning: elements of non-compliance detected!
1-5.sh	Warning: elements of non-compliance detected!
1-6.sh	Warning: elements of non-compliance detected!
2-1.sh	Warning: elements of non-compliance detected!
2-3.sh	Warning: elements of non-compliance detected!
3-1.sh	Warning: elements of non-compliance detected!
4-6.sh	Warning: elements of non-compliance detected!
4-9.sh	Warning: elements of non-compliance detected!
6-1-3.sh	Warning: elements of non-compliance detected!

**Figura 3.6:** Tabella del report HTML prodotta per il Progetto 1: nessuna raccomandazione è rispettata

di disporre di un dominio aziendale verificato da Google. Poiché l'obiettivo è semplicemente quello di utilizzare e validare il framework creato, il numero di risorse create è molto ridotto, e comunque distante dalle dimensioni di un'infrastruttura reale. Nello specifico, la Tabella 3.3 fornisce informazioni sulle risorse create. Ad esse, si aggiungono le decine di risorse create di default da Google per i fini più disparati, e.g. dischi, firewall, subnet, log bucket. Nel primo progetto, **tutte le opzioni** di configurazione sono state impostate **volutamente non conformi** alle raccomandazioni prese in esame. Nel secondo progetto, **soltanto un sottoinsieme** delle configurazioni è stato impostato (volutamente) come non conforme. Ciò ha consentito di valutare la capacità del framework di identificare le criticità esistenti, anche in scenari diversi. La Tabella 3.4 specifica, per ciascuna raccomandazione, se essa è rispettata nei due progetti, nonché lo script bash eventualmente utilizzato in Google Cloud CLI per l'introduzione degli elementi di non conformità all'interno della configurazione. Per i controlli cui è associata la dicitura *default*, non è stato necessario modificare alcuna configurazione, in quanto la configurazione di default, di per sé, non rispetta le raccomandazioni del CIS GCP Benchmark e richiede, dunque, di essere modificata. Come è evidente nelle Figure 3.6 e 3.7, i risultati prodotti dal framework sui due progetti sono coerenti con quanto indicato in Tabella 3.4.

**Script Execution Report**

Script	Status
1-1.sh	Warning: elements of non-compliance detected!
1-4.sh	Warning: elements of non-compliance detected!
1-5.sh	OK! All settings are compliant!
1-6.sh	Warning: elements of non-compliance detected!
2-1.sh	Warning: elements of non-compliance detected!
2-3.sh	OK! All settings are compliant!
3-1.sh	Warning: elements of non-compliance detected!
4-6.sh	OK! All settings are compliant!
4-9.sh	Warning: elements of non-compliance detected!
6-1-3.sh	OK! All settings are compliant!

**Figura 3.7:** Tabella del report HTML prodotta per il Progetto 2: le raccomandazioni 1.5, 2.3, 4.6 e 6.1.3 sono rispettate



---

### Istruzioni per l'utilizzo del framework

---

#### 4.1 Organizzazione della repository GitHub

La repository GitHub del presente progetto è disponibile a *questo link*. Nella root directory sono contenute la documentazione del progetto (*GCP\_SA\_report.pdf*), una presentazione digitale (*pitch.pptx*) e la cartella *src*, che racchiude, a sua volta, le seguenti sottocartelle:

- **setup**: contiene gli script utilizzati per la creazione e la configurazione dei due progetti GCP di cui alla Sezione 3.3, rispettivamente nelle sottocartelle *sample-env1* e *sample-env2*;
- **checks**: contiene gli script per l'esecuzione dei controlli.

#### 4.2 Utilizzo del framework

L'operazione da eseguire preliminarmente è quella di installare Google Cloud CLI: *qui* la guida ufficiale per Linux. Per utilizzare il framework sui progetti *pteh-02* e *pteh-04* di cui alla Sezione 3.3, bisogna ottenere l'autorizzazione d'accesso da parte dell'autore degli stessi. In alternativa, è possibile utilizzarlo su un qualunque altro progetto GCP. È possibile utilizzare gli script in */src/setup* per ricreare gli ambienti di cui alla Sezione 3.3. In tal caso, l'ordine consigliato per l'invocazione degli script è *create-resources.sh*, *create-user.sh*, *create-servaccounts.sh*, *create-sinks.sh*, *add-audit-config.sh*. Si consideri che anche le API per il setup dell'ambiente

```
francesco@LAPTOP-LV0QSRTF:~/Desktop/GCP-SecurityAssessment/scripts/checks$ gcloud auth login
Your browser has been opened to visit:
https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=YLINQW7kGdwBKjhj6E5c9ekRhnz5YQ3&access_type=offline&code_challenge=j8Nwlg21CbKJxKuui8Vxk6yNsMifCS3NZPBT7w_N2Mcs&code_challenge_method=S256
```

**Figura 4.1:** URL mostrato in console per il completamento dell'autenticazione

sono soggette, nella versione gratuita, alle limitazioni di rate. Una volta stabilito l'accesso al progetto, prima di poter invocare gli script per i controlli (in */src/checks*), è necessario eseguire in Google Cloud CLI le operazioni seguenti.

1. Installare le ulteriori dipendenze necessarie.

```
sudo apt install jq
```

2. Rimuovere eventuali account e/o progetti GCP precedentemente impostati.

```
gcloud config unset account
gcloud config unset project
```

3. Autenticarsi con un account che ha il permesso di accedere al progetto su cui si intende utilizzare il framework.

```
gcloud auth application-default login
```

All'esecuzione del comando, sarà mostrato a video un URL su cui cliccare per completare, tramite browser, l'autenticazione (vedi Figura 4.1).

4. Impostare il progetto su cui eseguire i controlli.

```
gcloud config set project <PROJECT-ID>
```

5. Impostare il progetto di cui si vuole consumare le quote al momento dell'invocazione delle API (stesso progetto indicato al punto precedente).

```
gcloud auth application-default set-quota-project <PROJECT-ID>
```

Al termine dei suddetti step, è possibile utilizzare gli script in */src/checks* per l'effettuazione dell'audit. In particolare, lo script *check-all.sh* richiede di specificare su linea di comando

l’ID del progetto GCP da analizzare e il dominio autorizzato; quest’ultima informazione è necessaria per eseguire il controllo 1.1, riferito all’eventuale presenza, all’interno del progetto, di utenti appartenenti a un dominio diverso da quello specificato. Al termine dell’esecuzione, lo script crea nella directory corrente le cartelle *logs* e *insights*, in cui vengono rispettivamente salvati i file di log e i report HTML prodotti.

## CAPITOLO 5

---

### Conclusioni

---

Il progetto è consistito nello sviluppo di un framework, basato su script bash, per il *Security Assessment* delle impostazioni di configurazione di un progetto Google Cloud Platform (GCP). Il meccanismo di audit è stato implementato tenendo conto di un sottoinsieme delle raccomandazioni contenute nel **CIS Benchmarks** per GCP.

#### 5.1 Sviluppi futuri

Si è inteso, con il presente progetto, fornire una **Proof of Concept** che fungesse da punto di partenza per lo sviluppo di strumenti da usare in produzione. Innanzitutto, sarebbe opportuno dotarsi delle API a pagamento di GCP, in modo da rimuovere le limitazioni esistenti sul rate delle richieste. Inoltre, un possibile sviluppo futuro è quello di implementare script di verifica per tutti i controlli del CIS Benchmarks, nonché di integrare raccomandazioni contenute in ulteriori standard e linee guida; solo a titolo esemplificativo, si considerino quelle riportate nella parte introduttiva del Capitolo 3. Ulteriori propositi sono quello di potenziare il meccanismo di reporting, mediante la produzione di summary e dati statistici più elaborati, e di implementare un motore di remediation che automatizzi la correzione degli errori di configurazione individuati.

---

## Bibliografia

---

- [1] A. Akinade, A. Ige, and A. Pub, "Cloud security challenges and solutions: A review of current best practices," *International Journal of Multidisciplinary Research and Growth Evaluation*, vol. 06, pp. 26–35, 12 2024. (Citato a pagina 1)
- [2] S. One. Google cloud platform (gcp) security checklist for 2025. [Online]. Available: <https://www.sentinelone.com/cybersecurity-101/cloud-security/gcp-security-checklist/> (Citato a pagina 1)
- [3] G. Cloud. Google cloud documentation. [Online]. Available: <https://cloud.google.com/docs/overview?hl=it> (Citato alle pagine 2 e 4)
- [4] C. for Internet Security (CIS). Cis benchmarks list. [Online]. Available: <https://www.cisecurity.org/cis-benchmarks> (Citato a pagina 11)