Cloud Security:
Securing The Invisible Thing

Mohammad Febri R, OSCP, CEH – Sr. Security Engineer
mohammad.ramadlan@tiket.com

Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation

Agenda

1. Introduction
2. Background
3. Objective
4. Cloud IAM
5. Result

1.

Introduction

About Me

- Mohammad Febri Ramadlan (Ebi)
- Information Security Consultant
- Open-source Enthusiasts (OWASP Project Leader and Mozilla Keyholder)
- Par-Time Blogger, Swimmer, and Musician

Contact:

- +6281809809636

2.

Background

Background

- PCI DSS: 18. Data Control & Access Control Policies
- ISO 27001: Annex A.9: Access Control

3.

Objective

Objective

- Fulfill the KPI
- Improve the current process
- User access monitoring
- Access control review

INDONESIA OpenInfra Days

4.

Cloud IAM

Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation

Cloud

Google Cloud Platform

Cloud IAM

“ who (identity) has what access (role) for which resource.”

IAM Entity

- User and Group
- Service Account
- Role
- Policy
- Version
- Environment

Tools Option

- *Forseti*
- *Security monkey*
- *Dollhouse*
- *ScoutSuite*
- *CloudSploit*

Sample:

ScoutSuite

```
def run_from_cli():
    parser = ScoutSuiteArgumentParser()
    args = parser.parse_args()

    # Get the dictionary to get None instead of a crash
    args = args.__dict__

    # TODO provider-specific arguments should be prepended with the provider's code
    #  (e.g. aws_profile, azure_user_account)

    try:
        return run(provider=args.get('provider'),
                   # AWS
                   profile=args.get('profile'),
                   aws_access_key_id=args.get('aws_access_key_id'),
                   aws_secret_access_key=args.get('aws_secret_access_key'),
                   aws_session_token=args.get('aws_session_token'),
                   # Azure
                   user_account=args.get('user_account'), service_account=args.get('service
_account'),
                   cli=args.get('cli'), msi=args.get('msi'), service_principal=args.get('se
rvice_principal'), file_auth=args.get('file_auth'),
                   tenant_id=args.get('tenant_id'), subscription_id=args.get('subscription_
id'),
                   client_id=args.get('client_id'), client_secret=args.get('client_secret')
,
                   username=args.get('username'), password=args.get('password'),
                   # GCP
                   project_id=args.get('project_id'), folder_id=args.get('folder_id'), orga
nization_id=args.get('organization_id'), all_projects=args.get('all_projects'),
                   # Aliyun
                   access_key_id=args.get('access_key_id'), access_key_secret=args.get('acc
ess_key_secret'),
```

Sample (2):

ScoutSuite Services

```
class GCPServicesConfig(BaseServicesConfig):
    def __init__(self, credentials=None, default_project_id=None,
                 project_id=None, folder_id=None, organization_id=None, all_projects=None,
                 **kwargs):

        super(GCPServicesConfig, self).__init__(credentials)

        facade = GCPFacade(default_project_id, project_id, folder_id, organization_id, all_
projects)

        self.cloudresourcemanager = CloudResourceManager(facade)
        self.cloudsql = CloudSQL(facade)
        self.cloudstorage = CloudStorage(facade)
        self.computeengine = ComputeEngine(facade)
        self.iam = IAM(facade)
        self.kms = KMS(facade)
        self.stackdriverlogging = StackdriverLogging(facade)

        # Instantiate proprietary services
        try:
            self.kubernetesengine = KubernetesEngine(facade)
        except NameError as _:
            pass

    def _is_provider(self, provider_name):
        return provider_name == 'gcp'
```

Sample (3):

ScoutSuite Cloudsql

```
    async def get_backups(self, project_id: str, instance_name: str):
        try:
            cloudsql_client = self._get_client()
            backups_group = cloudsql_client.backupRuns()
            request = backups_group.list(project=project_id, instance=instance_name)
            return await GCPFacadeUtils.get_all('items', request, backups_group)
        except Exception as e:
            print_exception('Failed to retrieve database instance backups: {}'.format(e))
            return []

    async def get_database_instances(self, project_id: str):
        try:
            cloudsql_client = self._get_client()
            instances_group = cloudsql_client.instances()
            request = instances_group.list(project=project_id)
            return await GCPFacadeUtils.get_all('items', request, instances_group)
        except Exception as e:
            print_exception('Failed to retrieve database instances: {}'.format(e))
            return []

    async def get_users(self, project_id: str, instance_name: str):
        try:
            cloudsql_client = self._get_client()
            response = await run_concurrently(
                    lambda: cloudsql_client.users().list(project=project_id, instance=ins
tance_name).execute()
            )
            return response.get('items', [])
        except Exception as e:
            print_exception('Failed to retrieve database instance users: {}'.format(e))
            return []
```

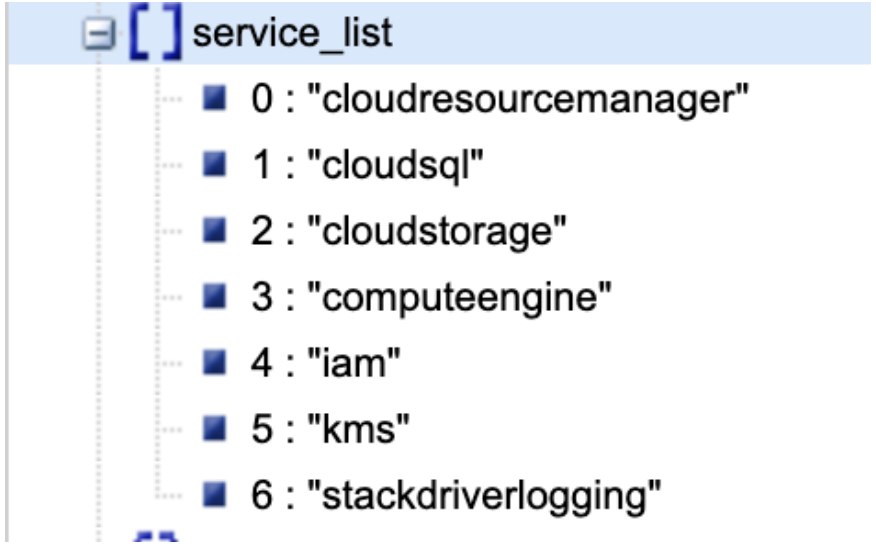
5.

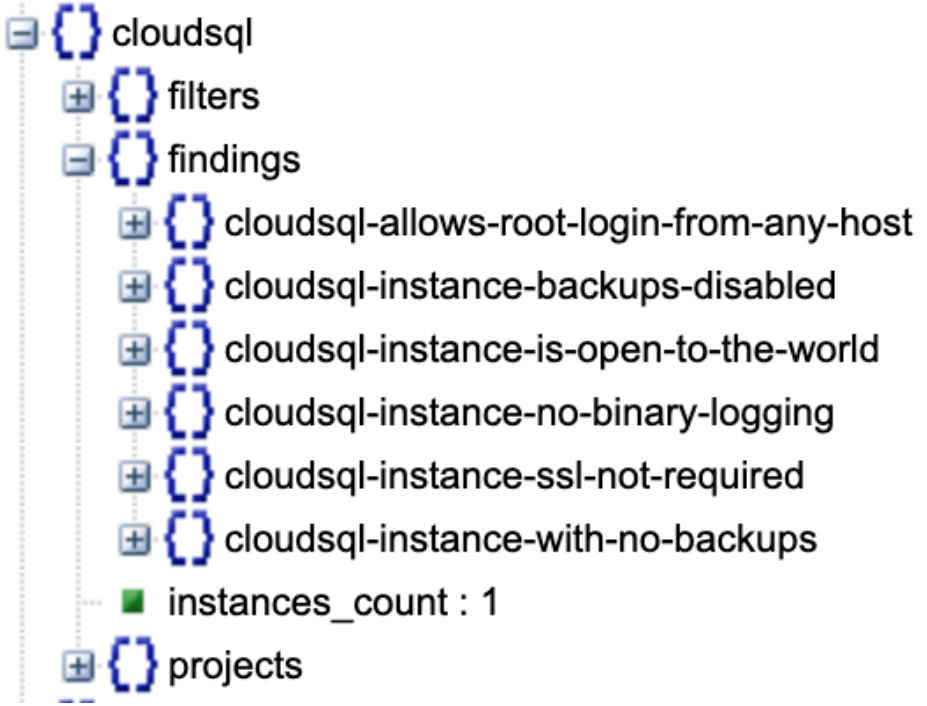
Result

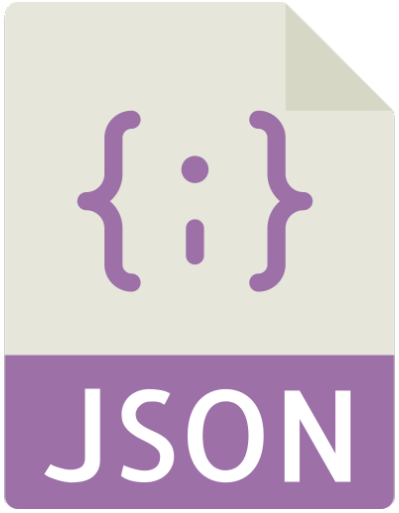
JSON File


```
{"account_id": "security-research-246306","all_projects": false,"environment": "default","folder_id": null,"last_run": {"ruleset_about": "This ruleset consists of nu
merous rules that are considered standard by NCC Group. The rules enabled range from violations of well-known security best practices to gaps resulting from less-kno
wn security implications of provider-specific mechanisms. Additional rules exist, some of them requiring extra-parameters to be configured, and some of them being ap
plicable to a limited number of users.","ruleset_name": "default","run_parameters": {"excluded_regions": null,"regions": null,"services": [],"skipped_services": []},
"summary": {"cloudresourcemanager": {"checked_items": 132,"flagged_items": 10,"max_level": "warning","resources_count": 39,"rules_count": 5},"cloudsql": {"checked_it
ems": 5,"flagged_items": 2,"max_level": "warning","resources_count": 1,"rules_count": 6},"cloudstorage": {"checked_items": 8,"flagged_items": 4,"max_level": "warning
","resources_count": 2,"rules_count": 4},"computeengine": {"checked_items": 230,"flagged_items": 49,"max_level": "warning","resources_count": 53,"rules_count": 11},"
iam": {"checked_items": 22,"flagged_items": 3,"max_level": "warning","resources_count": 5,"rules_count": 4},"kms": {"checked_items": 0,"flagged_items": 0,"max_level"
: "warning","resources_count": 0,"rules_count": 0},"stackdriverlogging": {"checked_items": 0,"flagged_items": 0,"max_level": "warning","resources_count": 0,"rules_co
unt": 1}},"time": "2019-08-21 09:28:56+0000","version": "5.3.2"},"metadata": {"compute": {"computeengine": {"resources": {"firewalls": {"cols": 2,"count": 22,"full_p
ath": "services.computeengine.projects.id.firewalls","path": "services.computeengine.projects.id.firewalls","script": "services.computeengine.projects.firewalls"},"i
nstances": {"cols": 2,"count": 7,"full_path": "services.computeengine.projects.id.zones.id.instances","path": "services.computeengine.projects.id.zones.id.instances"
,"script": "services.computeengine.projects.zones.instances"},"networks": {"cols": 2,"count": 2,"full_path": "services.computeengine.projects.id.networks","path": "s
ervices.computeengine.projects.id.networks","script": "services.computeengine.projects.networks"},"snapshots": {"cols": 2,"count": 0,"full_path": "services.computeen
gine.projects.id.snapshots","path": "services.computeengine.projects.id.snapshots","script": "services.computeengine.projects.snapshots"},"subnetworks": {"cols": 2,"
count": 22,"full_path": "services.computeengine.projects.id.regions.id.subnetworks","path": "services.computeengine.projects.id.regions.id.subnetworks","script": "se
rvices.computeengine.projects.regions.subnetworks"}}},"kubernetesengine": {"resources": {"clusters": {"cols": 2,"path": "services.kubernetesengine.projects.id.zones.
id.clusters"}}}},"database": {"cloudsql": {"resources": {"instances": {"cols": 2,"count": 1,"full_path": "services.cloudsql.projects.id.instances","path": "services.
cloudsql.projects.id.instances","script": "services.cloudsql.projects.instances"}}},"management": {"stackdriverlogging": {"resources": {"sinks": {"cols": 2,"count":
 0,"full_path": "services.stackdriverlogging.projects.id.sinks","path": "services.stackdriverlogging.projects.id.sinks","script": "services.stackdriverlogging.projec
ts.sinks"}}}},"security": {"cloudresourcemanager": {"resources": {"bindings": {"cols": 2,"count": 31,"full_path": "services.cloudresourcemanager.projects.id.bindings
","path": "services.cloudresourcemanager.projects.id.bindings","script": "services.cloudresourcemanager.projects.bindings"},"groups": {"cols": 2,"count": 0,"full_pat
h": "services.cloudresourcemanager.projects.id.groups","path": "services.cloudresourcemanager.projects.id.groups","script": "services.cloudresourcemanager.projects.g
roups"},"users": {"cols": 2,"count": 8,"full_path": "services.cloudresourcemanager.projects.id.users","path": "services.cloudresourcemanager.projects.id.users","scri
pt": "services.cloudresourcemanager.projects.users"}}},"iam": {"resources": {"service_accounts": {"cols": 2,"count": 5,"full_path": "services.iam.projects.id.service
_accounts","path": "services.iam.projects.id.service_accounts","script": "services.iam.projects.service_accounts"}}},"kms": {"resources": {"keyrings": {"cols": 2,"co
unt": 0,"full_path": "services.kms.projects.id.keyrings","path": "services.kms.projects.id.keyrings","script": "services.kms.projects.keyrings"}}}},"storage": {"clou
dstorage": {"resources": {"buckets": {"cols": 2,"count": 2,"full_path": "services.cloudstorage.projects.id.buckets","path": "services.cloudstorage.projects.id.bucket
s","script": "services.cloudstorage.projects.buckets"}}}}},"organization_id": null,"project_id": null,"provider_code": "gcp","provider_name": "Google Cloud Platform"
,"result_format": "json","service_list": ["cloudresourcemanager","cloudsql","cloudstorage","computeengine","iam","kms","stackdriverlogging"],"services": {"cloudresou
rcemanager": {"bindings_count": 31,"filters": {"cloudresourcemanager-gmail-accounts-used": {"checked_items": 8,"dashboard_name": "Users","description": "Gmail accoun
ts in use","flagged_items": 0,"id_suffix": "name","items": [],"level": "warning","path": "cloudresourcemanager.projects.id.users.id","rationale": "<b>Description:</b
><br><br>Gmail accounts are personally created and controllable accounts. Organizations seldom have any control over them. Thus, it is recommended that you use fully
 managed corporate Google accounts for increased visibility, auditing, and control over access to Cloud Platform resources.<br><br><b>References:</b><ul><li>CIS Goog
le Cloud Platform Foundations v1.0.0 1.1</li></ul>","service": "Cloud Resource Manager"},"cloudresourcemanager-primitive-role-in-use": {"checked_items": 31,"dashboar
d_name": "Bindings","description": "Primitive Role In Use","flagged_items": 3,"id_suffix": "this","items": ["cloudresourcemanager.projects.security-research-246306.b
indings.96788b38514f6e70fbc596eb5bf118cc55f69f01.this","cloudresourcemanager.projects.security-research-246306.bindings.ebc06726978fbbb8850f2631ec6aae9bfebd7b13.this
","cloudresourcemanager.projects.security-research-246306.bindings.2920a274ea424493e0f6ba82c83caad928cc3060.this"],"level": "warning","path": "cloudresourcemanager.p
rojects.id.bindings.id","rationale": "No description available.","service": "Cloud Resource Manager"},"cloudresourcemanager-role-assigned-to-user": {"checked_items":
 31,"dashboard_name": "Bindings","description": "IAM Role Assigned to User","flagged_items": 5,"id_suffix": "users","items": ["cloudresourcemanager.projects.security
-research-246306.bindings.341a02697dc953f6c3b95c0cb2149e24249e3a69.users","cloudresourcemanager.projects.security-research-246306.bindings.3379db720d58c9012e6e2a5bff
dda4149c8aeef3.users","cloudresourcemanager.projects.security-research-246306.bindings.ebc06726978fbbb8850f2631ec6aae9bfebd7b13.users","cloudresourcemanager.projects
.security-research-246306.bindings.cd9fac626aded87c804ed7b6313c85b4365731f0.users","cloudresourcemanager.projects.security-research-246306.bindings.2920a274ea424493e
0f6ba82c83caad928cc3060.users"],"level": "warning","path": "cloudresourcemanager.projects.id.bindings.id","rationale": "<b>Description:</b><br><br>Best practices rec
ommends granting roles to a Google Suite group instead of to individual users when possible. It is easier to add members to and remove members from a group instead o
f updating a Cloud IAM policy to add or remove users.","service": "Cloud Resource Manager"},"cloudresourcemanager-sa-has-admin-privileges": {"checked_items": 31,"das
hboard_name": "Bindings","description": "Service Account with Admin Privileges","flagged_items": 1,"id_suffix": "service_accounts","items": ["cloudresourcemanager.pr
```

JSON View


service_list
0 : "cloudresourcemanager"
1 : "cloudsql"
2 : "cloudstorage"
3 : "computeengine"
4 : "iam"
5 : "kms"
6 : "stackdriverlogging"


cloudsql
filters
findings
cloudsql-allows-root-login-from-any-host
cloudsql-instance-backups-disabled
cloudsql-instance-is-open-to-the-world
cloudsql-instance-no-binary-logging
cloudsql-instance-ssl-not-required
cloudsql-instance-with-no-backups
instances_count : 1
projects


JSON

Slack Notification: Services

[+] Service : cloudsql
[+] Data : findings
*** Dashboard : Instances
*** Description : Instance with automatic backups disabled
*** Rationale : Description:

Automatic backups should be configured for Cloud SQL instances in order to ensure backups are created regularly.
*** Dashboard : Instances
*** Description : Instance allows root login from any host
*** Rationale : Description:

Root access to MySQL Database Instances should be allowed only through trusted IPs.

References:
CIS Google Cloud Platform Foundations v1.0.0 6.4

Slack Notification: User and IAM Role

[+] User Count : 5
-----[+] Email : abu.halid@company.com
-----[+] Roles : [u'owner']
-----[+] Email : indah.sherly@company.com
-----[+] Roles : [u'owner']
-----[+] Email : test@company.com
-----[+] Roles : [u'owner', u'viewer']
-----[+] Email : jim.brian@company.com
-----[+] Roles : [u'owner']
-----[+] Email : mohammad.febri@company.com
-----[+] Roles : [u'iam.securityReviewer', u'viewer']

INDONESIA OpenInfra Days

Question

Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation

Summarize

1. Fulfill the regulation (PCI DSS & ISO 27001)
2. Cloud Audit is developed to ensure the proper user access
3. User access matrix review in daily activity

INDONESIA
OpenInfra Days

Thank you!

Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation