
# SANS SEC510 - Public Cloud Security



Ivan Vlad S. 2022

# Aa

**Access Token (JWT) Function Request** *[b4/p24]*

**Account Setup: AWS** *[b1/p30]* Follow page steps

**Account Setup: Azure** *[b1/p32]* Follow page steps

**Account Setup: GCP** *[b1/p33]* Follow page steps

**Active Directory (Azure AD)** *[b1/p105]* Azure AD is not the same and does not replace normal AD - not interchangeable. Azure AD = Azure's IAM. Can be sync'd with AD so users can use same creds for every app

**AD - Registering a Web App: Azure** *[b5/p31]* Terraform - register within Azure AD tenant

**AD Business-to-Customer B2C: Azure** *[b5/p46]* Similar to Cognito

**AD Service Principal - Creation with Terraform: Azure** *[b5/p15]* Terraform - each of these resources have to be managed independently

**AdministratorAccess Policy: AWS** *[b1/p84]* grants user unconditional ability to take any action on any resource in any service

**Advanced remote Access Assessment Criteria** *[b2/p118]* Require multiple factors of authentication for remote admin access

**Alibaba Cloud** *[b5/p98]* Next big 3 - Alibaba, IBM, Oracle; still niche players by Gartner

**Application Security Groups (ASG)** *[b2/p40]* group virtual machines, eliminate pinning to dynamic VM IPs

**App Engine - Custom Firewall Rules** *[b4/p80]* Supports several different firewall options for allowing or denying requests to running services: Network Sources, GCP Internal Sources, Internal and Cloud Load Balancing

**App Engine - Default Firewall Rule: GCP** *[b4/p79]* Default firewall rule allows **all** traffic (HTTP and HTTPS) from **all** IP address ranges out of the box, **implicit allow**

**App Engine - Default Service Account: GCP** *[b4/p82]* All App Engine services in a GCP project run under the same service account identity - WARNING: deleting the default service account will break current and future apps in Cloud project

**App Engine - Firewall Rule Hardening: GCP** *[b4/p81]* Example showing how to modify App Engine firewall rules to block all traffic originating from outside a trusted corp network

**App Engine - Flexible Instance Debugging: GCP** *[b4/p78]* Two paths for debugging: Enable Debug and SSH Access

**App Engine - Insecure Transport Hardening: GCP** *[b4/p84]* Benchmark 4.10 - ensure App Engine apps enforce HTTPS connections

**App Engine - Service Account Hardening: GCP** *[b4/p83]* Example for options for least privilege for App Engine default SA and provide long-lived credentials

**App Engine - User Authentication: GCP** *[b4/p86]* Provide several options to authenticate before accessing protected resources: Google Sign-in, OAuth 2.0 and OpenID connect, Firebase authentication

**App Engine - Web Security Scanner Integration: GCP** *[b4/p85]* Native integration with Cloud Web Security Scanner for dynamic scanning

**App Engine Environments: GCP** *[b4/p77]* Supports two runtime environment options: Standard Environment and Flexible Environment

**App Engine: GCP** *[b4/p76]* Provides balance between traditional app architectures requiring long-running web app processes, and the desire to eliminate responsibility for managing the underlying infra

**App Service - Admin Access: Azure** *[b4/p69]* Support managed SSH, browser-based console for Linux and Windows, connect to SSH from CLI for Linux

**App Service - App Service Plans with Functions: Azure** *[b4/p75]* Function Apps can be run on App Service Plan, WARNING - exposes functions to the same security concerns as long-running App Services

**App Service - FTP Deployments: Azure** *[b4/p73]* Benchmark 9.10 - ensure **FTP** deployments are **disabled**, even **FTPS**

**App Service - Insecure SSH Ciphers: Azure** *[b4/p71]* Must use cipher block chaining (CBC) to connect SSH, not considered secure by Microsoft

**App Service - Insecure Transport Handling: Azure** *[b4/p72]* Benchmark 9.2 - ensure webapp redirects all HTTP traffic to HTTPS in Azure App Service | Benchmark 9.3 - ensure webapp is using latest version of TLS

**App Service - Miscellaneous Benchmarks: Azure** *[b4/p74]* 9.1, 9.4, 9.5, 9.6-10

**App Service Assessment Criteria** *[b4/p66]* While app services abstract away from infra, they do NOT eliminate the associated security concerns - examine them

**App Service: Azure** *[b4/p68]* Create to host long-running applications

**AWS Benefits** *[b1/p7]* OG to market, lots of services, popularity, considered default choice

**AWS Config subscription** [b5/p56]

**AWS Security Benchmark - Open Source** *[b5/p51]* OG open-source tool for auditing AWS accounts against CIS benchmarks

**AWS Security Hub** *[b5/p52]* Key Features, uses **ASFF** for findings

**AWS Security Hub - Benchmark Standards** *[b5/p54]* Supports CIS, AWS Foundation Security Best Practices, PCI DSS

**AWS Security Hub - Config** *[b5/p53]* Terraform config

**AWS Security Hub - Custom Actions** *[b5/p62]* Terraform - create new custom action

**AWS Security Hub - Finding Disposition** *[b5/p61]* Each finding can be given disposition status - new, notified, suppressed, or resolved

**AWS Security Hub - Finding List** *[b5/p60]* findings filtered where "product name" is "Prowler"

**AWS Security Hub - Product Integration Example** *[b5/p59]* Terraform showing config subscribing to Prowler integration; Must ensure S Hub is configured and active in current region

**AWS Security Hub - Product Integrations** *[b5/p57]* allow native and 3rd party integrations

**AWS Security Hub - Security Findings Format** *[b5/p58]* ASFF - AWS Security Finding Format, slide shows abbreviated example of custom finding from Prowler

**AWS Security Hub - Standard Subscription Example** *[b5/p56]* Terraform - subscribe to Security Hub AWS Foundational Security Best Practices Benchmark

**Azure Benefits** *[b1/p9]* Competitive pricing, integrated with Microsoft services, won JEDI in 2019 - DoD canceled JEDI in 2021, Azure & AWS front-runners for new contract

**Azure Scout Suite Provider - Open Source** *[b5/p63]* Scout Suite gathers resources from each CSP for evaluation

**Azure Security Center** *[b5/p64]* Summarizes key features from: Cloud Inventory, Secure Score, Regulatory Compliance, Azure Defender

**Azure Security Center - Agent Provisioning** *[b5/p76]* Terraform, To collect data, Log Analytics Agent and Security Extensions are required

**Azure Security Center - Azure Defender** *[b5/p71]* Provides threat detection capabilities

**Azure Security Center - Enabling Azure Defender** *[b5/p73]* Terraform to enable

**Azure Security Center - Threat Detection** *[b5/p75]* Provided by Azure Defender: Azure Network Layer, Azure Resource Manager

**Azure Security Center - Workflow Automation** *[b5/p77]* invoke an Azure Logic App based on the item's data type, alert name, and severity

**Azure Security Center - Hybrid Cloud Protection** *[b5/p74]* Supports non-azure machines local or in other cloud (AWS, GCP) to be monitored

**Azure Security Center - Inventory** *[b5/p66]* Dashboard shows Total resources, Unhealthy resources, Unmonitored resources

**Azure Security Center - Pricing Models** *[b5/p65]* Free and Azure Defender

**Azure Security Center - Recommendation Details** *[b5/p69]* automated by combining commands 1 + 2

**Azure Security Center - Regulatory Compliance Policies** *[b5/p70]* Built-in policies and dash for PCI DSS, CIS, HIPAA HIRTUST, ISO 27001, NIST, SOC TSP, etc.

**Azure Security Center - Secure Score Dashboard** *[b5/p67]* shows overall compliance score for resources, not accessible via AZ CLI

**Azure Security Center - Secure Score Recommendations** *[b5/p68]* pass/fail score - can use AZ command: az security task list

# Bb

**Benchmark 1.19** *[b1/p82]* requires all AWS virtual machines to use instance profile roles for managing credentials

**Benchmark 1.22** *[b1/p85]* Ensure IAM policies that allow full *.* admin privileges are not created

**Benchmark 2.7** *[b3/p45]* ensure CloudTrail logs encrypted at rest using KMS CMKs

**Benchmark 2.9** *[b2/p65]* Ensure VPC flow logging is enabled on all VPCs

**Benchmark 3 Networking: 3.1, 3.6, 3.7** *[b2/p47]* Benchmark 3 Networking: 3.1 - ensure the default network does not exist in a project, 3.6 - ensure that SSH access is restricted from the internet, 3.7 - ensure RDP is restricted from internet

**Benchmark 3.9** *[b2/p81]* Ensure VPC flow logs is enabled for every subnet in VPC network

**Benchmark 4.10** *[b4/p84]* Benchmark 4.10 - ensure App Engine apps enforce HTTPS connections

**Benchmark 4.3** *[b2/p19]* Ensure the default security group of every VPC restricts all traffic

**Benchmark 4.9, 4.10, 4.11, 4.13** *[b3/p50]* Benchmark 4.9 - ensure data encryption is set On for SQL db, Benchmark 4.10 - ensure SQL server TDE protector is encrypted with BYOK, Benchmark 4.11 - Ensure enforce SSL is set to enabled for MySQL db, Benchmark 4.13 - Ensure enforce SSL is enabled for PostgreSQL db

**Benchmark 6.4** *[b2/p74]* Network security group flow logs should be enabled, and the retention period is set to greater than or equal to 90 days

**Benchmark 6: 6.1, 6.2, 6.3** *[b2/p37]* 6.1-ensure RDP restricted from internet, 6.2-ensure SSH "", 6.3-ensure SQL DBs do not allow ingress 0.0.0.0/0

**Benchmark 9.10** *[b4/p73]* Benchmark 9.10 - ensure FTP deployments are disabled

**Benchmark 9.2, 9.3** *[b4/p72]* Benchmark 9.2 - ensure webapp redirects all HTTP traffic to HTTPS in Azure App Service | Benchmark 9.3 - ensure webapp is using latest version of TLS

**Benchmarks 9.1, 9.4, 9.5, 9.6-10** *[b4/p74]* App Service - Miscellaneous Benchmarks: Azure

**BigQuery - Sharing Datasets** *[b3/p125]* Managed data warehouse - shows how to share a BQ dataset publicly

**Binding a Role at the Resource Level: GCP** *[b1/p140]* The better alternative to restricting permissions to specific resources is to bind a role with those permissions at the resource level

**Bucket Hardening - Customer Managed Encryption: GCS** *[b3/p102]* Terraform - set bucket's encryption option to a customer managed encryption key

**Bucket Hardening - Customer Managed Key Permissions: Azure** *[b3/p89]* Terraform - create an access policy for the storage accounts customer managed key

**Bucket Hardening - Customer Managed Key Permissions: GCS** *[b3/p103]* Terraform - grant the project's GCS service account permission to use the KMS key

**Bucket Hardening - Data Retention Policy: GCS** *[b3/p105]* Ensure integrity of audit data by ensuring GCS buckets config a retention policy with Retention Period and Bucket Lock

**Bucket Hardening - Data Retention: GCS** *[b3/p104]* Buckets storing sensitive audit records must enable several key features to ensure objects are retained: Data Retention & Lifecycle Requirements

**Bucket Hardening - Object Lifecycle: GCS** *[b3/p106]* Object lifecycle ruleset for transferring audit data between storage classes and eventually deleting records after retention period has been met

**Bucket Hardening - Uniform Access Control & Logging: GCS GCP** *[b3/p101]* Terraform config for bucket to use the Uniform Access Control permissions

**Built-in Owner Role Definition: Azure** *[b1/p117]* Azure definitions are defined in JSON, key settings: AssignableScopes, Actions, Miscellaneous Properties

**Built-in Reader and Data Access Role: Azure** *[b1/p118]* Azure built-in roles are convenient because the role definition maintenance falls into the CSP for shared responsibility

**Built-in Role Definitions: Azure** *[b1/p116]* Azure managed for customers to use, include Owner, Contributor, Storage Blob Data Contributor

**BYOK Azure DB for MsSQL** *[b3/p54]* Encryption Required, BYOK **NOT** supported

# Cc

**Capital One Breach: Credential Mgmt Gone Wrong** *[b1/p49]* 2019 - Cap One breached by SSRF attack.

**cheetah.go, cheetah.yaml** *[b4/p33]* secrets mgmt. review

**CIS Cloud Provider Benchmarks** *[b1/p20]* CSP assessment checklist, limited but provides foundational baseline for key CSP services

**Classic VPN Gateway Example: GCP** *[b2/p142]* Supports single external IP and single tunnel

**Cloud VPN: GCP** *[b2/p141]* Provides IPsec VPN connectivity, classic VPN, HA VPN

**Client VPN Authorization Rule: AWS** *[b2/p127]* Terraform - create client VPN authorization rule allowing access to a VPC network

**Client VPN Config: AWS** *[b2/p125]* client VPN config uses mutual certification authentication for access control

**Client VPN Example: AWS** *[b2/p126]* Create AWS Client VPN Endpoint using terraform

**Cloud Advanced Remote Access Services** *[b2/p119]* AWS Session Manger / VPN, Azure Serial Console / VPN Gateway, GCP Cloud SSH / Cloud VPN

**Cloud Application Services** *[b4/p65]* AWS Elastic Beanstalk, Azure App Service, GCP App Engine

**Cloud Compliance Services** *[b5/p50]* AWS Security Hub, Azure Security Center, GCP Security Command Center

**Cloud Compliance Services Summary** *[b5/p90]* Comparing the Cloud Compliance Service Platforms: Managed Benchmarks, 3P Integrations, Custom Findings, Event Hooks, Multicloud

**Cloud End-to-End Encryption** *[b3/p42]* Secure communication to the cloud and communications WITHIN the cloud

**Cloud End-User Identity Solutions** *[b5/p39]* AWS Cognito User Pools, Microsoft Identity Platform and Azure AD B2C, Google CICP and Firebase Authentication

**Cloud Instance Metadata API** *[b1/p45]* MITRE ATT&CK T1522: service acc creds and config data stored in Instance Metadata Service (IMDS)

**Cloud Key Management Systems** *[b3/p5]* AWS KMS, Azure Key Vault, Google Cloud KMS

**Cloud Private Service Endpoint Benefits** *[b2/p97]* Keeps all resources in VPC private, able to connect to cloud services without exposing traffic to the internet - no NAT or IG required

**Cloud Private Service Endpoints** *[b2/p96]* Create a private link between VPC resources and cloud services (no internet traffic)

**Cloud Resource Hijacking** *[b1/p77]* MITRE ATT&CK T1496: consuming victim's cloud resources to solve resource intensive problems

**Cloud Serverless Assessment Criteria** *[b4/p10]* Review serverless functions and ensure you are doing the following

**Cloud Serverless Event Driven Architecture** *[b4/p5]* HTTPS/API Gateway > SDK > Scheduled Events

**Cloud Serverless Execution Environment** *[b4/p7]* Difficult to assess because infra and container runtime is managed by CSP (inside underlying platform)

**Cloud Serverless Execution Environments** *[b4/p9]* Each functions execution environment: Runtime, OS, Default Directory, User

**Cloud Serverless Platforms** *[b4/p3]* AWS Lambda, Azure Functions, GCP Cloud Functions

**Cloud Serverless Reverse Engineering - Serverless Prey** *[b4/p8]* Open-source project for reverse engineering serverless execution environments across Big 3, contains IaC (Terraform and Serverless Framework) for deploying 3 different functions: Panther, Cougar, Cheetah

**Cloud Service Discovery** *[b2/p94]* MITRE ATT&CK T1526 - enumerating the cloud services accessed by a system after gaining access (used by a CSC)

**Cloud Services: Early** *[b1/p39]* Originally limited, AWS S3, SQS (2006), Microsoft SQL offering (2009), Google App Engine (2008)

**Cloud Services: Today** *[b1/p40]* Service for everything now, access mgmt is critical

**Cloud Single Sign-On Solutions** *[b5/p23]* AWS SSO, Microsoft Identity Platform & Azure AD, Google Cloud Identity

**Cloud Storage Platforms** *[b3/p64]* AWS S3, Azure Storage, GCP Cloud Storage

**Cloud Storage Security Controls** *[b3/p66]* Public access config options, least priv / RBAC, signed URL object sharing, versioning, retention lifecycle policies, monitoring & logging, secure data transport

**Cloud Virtual Network Monitoring** *[b2/p64]* Starts with capturing info about traffic flow: AWS VPC Flow Logs, Azure NSG Flow Logs, GCP VCP Flow Logs

**Cloud Virtual Network Security Controls** *[b2/p8]* Key controls: default network config, traffic flow/firewall rules, virtual network traffic monitoring, private endpoint security, VPN gateway options

**Cloud Virtual Network Services** *[b2/p7]* AWS VPC, Azure VNet, GCP VPC

**Cognito User Pools** *[b5/p41]* Unique feature is Hosted UI, create dedicated accounts for an app with username and pass

**Cognito User Pools - Configuring a Web App** *[b5/p44]* Node.js app for Cognito auth

**Cognito User Pools - Creating a User Pool** *[b5/p42]* Terraform creating user pool

**Cognito User Pools - Creating a User Pool Client** *[b5/p43]* Terraform - creating authorized client app for Cognito auth

**Cognito User Pools - Sample ID Token** *[b5/p45]* decoded JWT example

**Condition Example: AWS IAM** *[b1/p89]* use condition statement to add more restriction to policy

**Conditional Access Policies: Azure** *[b1/p123]* Policies applied after first-factor of authentication: Signal, Assignment, Access Controls

**Container_acces_type = "container"** | Anonymous read access to the container

**Credential Management Assessment Criteria** *[b1/p67]* Configure IMDS to be as inaccessible as possible

**Credential Mgmt Gone Wrong: Evil Request** *[b1/p52]* example shows malicious request to AWS IMDS

**Credential Mgmt Gone Wrong: Normal Request** *[b1/p51]* example is vulnerable to SSRF

**Credential Mgmt Gone Wrong: SSRF** *[b1/p50]* occur when an app requests data from another URL supplied from an untrusted location - can allow unauthorized access

**Credential Pivoting: AWS** *[b1/p62]* set access key ID, secret access key, session token environment variables to the stolen creds

**Credential Pivoting: Azure** *[b1/p63]* Set stolen JWT to environment variable, match auth header, submit request to API

**Credential Pivoting: GCP** *[b1/p65]* Set stolen token to environment variable, match auth header, submit request to API

**Cross-Cloud Access Management Considerations** *[b5/p8]* Not as secure as authorizing to single cloud, proper key access mgmt for long-lived creds is even more important

**Cryptographic Key Management Assessment Criteria** *[b3/p3]* Limit and Audit all cryptographic key usage – use **Soft** deletion method

**Cryptographic Key** | lost key would impact **availability**

**curl -s "http.."** *[b1/46]* metadata service example

**curl -H "Secret.."** [b4/p24] JWT valid for 8 hours

**Custom Network Controls: AWS** *[b2/p20]* Create custom VPC resources to enable controls: NAT/egress only gateway, private subnets, ingress and egress traffic filtering

**Custom Roles: GCP** *[b1/p132]* Enables you to enforce least privilege

# Dd

**Data Encryption - Azure DB for MySQL, PGSQL, and Maria** *[b3/p54]* Encryption through Azure Storage Service is always on, AES 256, can't use custom encryption key

**Data Encryption - Google Cloud SQL** *[b3/p57]* Most consistent manage db service of CSPs, single service for RDBs: Cloud SQL, which supports MySQL, PostgreSQL, and Microsoft SQL

**Data Encryption Assessment Criteria** *[b3/p43]* All data should be encrypted at rest and in-transit (extremely few exceptions)

**Data Encryption Assessment Criteria: Azure Database Service** *[b3/p50]* Benchmark 4.9 - ensure data encryption is set On for SQL db, Benchmark 4.10 - ensure SQL server TDE protector is encrypted with BYOK, Benchmark 4.11 - Ensure enforce SSL is set to enabled for MySQL db, Benchmark 4.13 - Ensure enforce SSL is enabled for PostgreSQL db

**Data Exfiltration Azure** *[b3/p118]* snapshots can't be shared permanently

**Data Exfiltration Paths** *[b3/p111]* Resources can be made public using two different methods: Sharing API, Resource Policy

**Data from Cloud Storage Object** *[b3/p65]* MITRE ATT&CK T1530: improperly secured cloud storage object

**Database Firewall Rule: Azure** *[b2/p41]* Terraform example of MySQL ingress

**Default Network ACL: AWS** *[b2/p14]* Exist in every region, ingress and egress rules allows all traffic on all ports

**Default Network Firewall rules: GCP VPC** *[b2/p45]* Table showing open admin access of default firewall

**Default Security Groups: AWS** *[b2/p16]* Exist in every region, ingress rules allow all traffic from associated instances, egress rule allows all traffic on all ports

**Default Virtual Machine Network Access: GPC VPC** *[b2/p46]* VMs created in console deploy into default VPC network, no warnings about wide-open SSH and RDP access

**Default Virtual Machine Network: AWS** *[b2/p17]* EC2 created in console/UI use default VPC network: wide open ACL rules, auto assigned public IP address

**Default Virtual Machine Security Group: AWS** *[b2/p18]* Ec2 created in console create security group with default open admin access: linux auto populate open SSH access, windows auto populate RDP access

**Default VPC Assessment Criteria: AWS** *[b2/p19]* Ensure the default security group of every VPC restricts all traffic

**Default VPC Hardening - Terraform: GCP** *[b2/p49]* Terraform run shell commands through a "null resource" example - orchestrate gcloud CLI commands

**Default VPC Hardening: GCP** *[b2/p48]* Commands to delete default firewall rules and VPC from GCP CLI

**Default VPC: AWS** *[b2/p11]* New accounts contain a default VPC in each region

**Default VPC: GCP VPC** *[b2/p44]* Contain 1 subnet for each region, default firewall rules are more permissive than AWS and Azure

**Disk Level Encryption BYOK: Azure SQL (MSSQL)** *[b3/p52]* Azure CLI create new key in vault (no way to configure TDE using Terraform)

**Disk Level Encryption: AWS RDS** *[b3/p46]* Terraform - encrypt db storage volume with KMS key

**Disk Level Encryption: Azure SQL (MSSQL)** *[b3/p51]* TDE - transparent data encryption, data and log files are encrypted and decrypted in real-time

# Ee

**EC2 - Sharing Machine Images and Disk Snapshots: AWS** *[b3/p113]* Sharing the image should be monitored closely to ensure attackers are not providing a path to exfiltrate data

**Effective Permissions** *[b1/96]* identity, resource, session

**Elastic Beanstalk: AWS** *[b4/p67]* Designed to simplify app deployment process while allowing customer to access underlying infra used in EC2

**Encryption at Rest - Disk Level** *[b3/p38]* Protecting data where it is STORED, entire storage medium is encrypted - protects against stolen disks and audits for encryption at rest

**Encryption at Rest - Record Level** *[b3/p39]* Individual records are encrypted separately, success comes down to key mgmt, only few technologies that enable record-level encryption

**Encryption at Rest: AWS CloudTrail** *[b3/p45]* Benchmark 2.7 - ensure CloudTrail logs encrypted at rest using KMS CMKs

**Encryption In-transit** *[b3/p41]* TLS, cloud providers allow TLS by default but also allow insecure connections by default

**Endpoint Network Access Validation: Azure** *[b2/p112]* Azure Key Vault inaccessible, shows endpoint control in action

**env command** | returns function metadata

**Envelope Encryption** | encrypting data with data key > encrypt data key with CMK

# Ff

**firebaseio.com/. json** *[b4/p109]* access Realtime db

**Firewall Egress Rule: GCP** *[b2/p56]* Example definition of an Egress firewall rule

**Firewall Implied Rules: GCP** *[b2/p51]* Every VPC has 2 implied firewall rules that are not visible and cannot be removed

**Firewall Ingress Rule: GCP** *[b2/p55]* Example definition of an ingress firewall rule

**Firewall Rule Components: GCP VPC** *[b2/p43]* Global network firewall, apply STATEFUL rules to ALL instances running in VPC network

**Firewall Rule Network Tags: GCP** *[b2/p53]* Arbitrary attributes for applying traffic flow rules to clusters of VMs with the same tag value

**Firewall Rule Service Account Targets: GCP** *[b2/p54]* Apply traffic flow rules to clusters of instances running the same service account

**Firewall Rule Targets: GCP** *[b2/p52]* Apply traffic flow control to one or more VMs for each firewall rule

**Firewall Rules - Retrieve Google's IP Space: GCP** *[b2/p140]* Bash script to retrieve Google's internal IP address ranges from their public SPF records automatically

**Firewall** | NSG & Azure Firewall are both **STATEFUL**

**Free Trials for CSP** *[b1/p29]* AWS, Azure = 12mo's free tier, GCP 12mo's for $300

**Function - Application Insights Telemetry: Azure** *[b4/p31]* Terraform - resource capturing function telemetry | By default, not enabled - must enable integration with Azure's Application Insight service

**Functions - Authenticating Users: GCP** *[b4/p41]* Use one of the following options for granting permissions to invoke a function: Function-function Access, End user function access, Google cloud IAM

**Functions - Authorization Level in functions.json: Azure** *[b4/p29]* Example of config for anonymously accessible function

**Functions - Default Network Config: Azure** *[b4/p25]* Natively support HTTPS triggers for invoking functions

**Functions - Default Network Config: GCP** *[b4/p36]* Naively support HTTPS triggers for invoking functions

**Functions - Environmental Variables: GCP** *[b4/p34]* Serverless Prey reverse shell running the env command to view the Cheetah function's environment variables

**Functions - Function Identity: GCP** *[b4/p40]* Editor role is too permissive for what your function needs in production

**Functions - HTTPS Trigger Authorization Level: Azure** *[b4/p28]* Example C# function's Http Trigger config - options available: Anonymous, Function, Admin

**Functions - IMDS Hardening: GCP** *[b4/p38]* Serverless Framework - setting environment variable

**Functions - Managed Identity Permissions: Azure** *[b4/p30]* Terraform - Function's role definition and scope for least privilege

**Functions - Managing Access: GCP** *[b4/p39]* Cloud Functions automatically create HTTPS trigger with a public IP address

**Functions - Require SSL/TLS: Azure** *[b4/p27]* Terraform - required fields to create a Function App

**Functions - Service Account Credentials: GCP** *[b4/p35]* Cloud Functions run under App Engine's default service account, which has Editor role on the project - dangerous permission level

**Functions - Source Code: GCP** *[b4/p33]* live in /srv/files directory

**Functions - Virtual Network Integration: Azure** *[b4/p32]* VNet integration requires Standard or Premium plan

**Functions - VPC Service Controls: GCP** *[b4/p42]* Cloud Functions support integration with VPC Service Controls - must be configured by an Organization Manager

**Functions Environment - Environment Variables: AWS** *[b4/p23]* Serverless Prey reverse shell running the end command to view the Cougar function's environment tables

**Functions Environment - Managed Credentials: Azure** *[b4/p24]* Compared to Lambda, Azure Functions are stored more securely.

**Functions Environment - Persistence: Azure** *[b4/p54]* Runtime environments contain several writable directories

**Functions Environment - Persistence: GCP** *[b4/p55]* Runtime environments contain several writable directories

**Functions Environment - Source Code: Azure** *[b4/p22]* lives in/home/site/wwwroot directory

**Functions Security Controls: Azure** *[b4/p26]* Modify default configs to harden Azure Function environment

**Functions Security Controls: GCP** *[b4/p37]* Hardening Google Cloud Functions environment

**Functions: Azure** *[b4/p20]* Like AWS Lambda, built on App Service Plan, Azure Storage, Managed Identity

# Gg

**G Suite: Identity in GCP** *[b1/p125]* G Suite is collection of Google product offerings, has Identity for Users and Groups in GCP

**Gartner MQ for Cloud** *[b1/p5]* 2014 to 2019 less providers, big 3 still the same and growing

**GCP Audit Logs - GCS Data Access Logging:** *[b3/p100]* Terraform config for data access audit logs for all Google services

**GCP Benefits** *[b1/p11]* competitive pricing, very diff from AWS, open-source tech: K8s, TensorFlow, Unique offerings: Firebase and Stackdriver

**GCP Security Command Center** *[b5/p80]* (SCC) provides intelligence across entire GCP Org

**GCP Security Command Center - Container Threat Detection** *[b5/p86]* Premium feature - monitors container images and runtime for attacks

**GCP Security Command Center - Dashboard** *[b5/p84]* Can toggle between active and inactive state for suppressing false positives or risk accepting known issues

**GCP Security Command Center - Event Threat Detection** *[b5/p85]* Premium feature - parses log sources for malicious activity

**GCP Security Command Center - Pricing Tiers** *[b5/p81]* Standard Tier (free) and Premium Tier

**GCP Security Command Center - Regulatory Compliance Policies** *[b5/p83]* Supports CIS GCP Foundation Benchmarks, PCI DSS, NIST 800-53, ISO 27001

**GCP Security Command Center - Security Health Analytics** *[b5/p82]* The key threat prevention feature, performs batch scan 2x a day every 12 hrs

**GCP Security Command Center - Web Security Scanner** *[b5/p87]* Premium feature - provides DAST for web apps

**GCP VPN Gateway Client/Point-to-site options** *[b2/p144]* GCP VPN does not support point-to-site connections

**Google Cloud Identity** *[b5/p37]* CICP - cloud identity for customers and partners; manage users in org and org's end-users

**Google Cloud Identity - Managed vs Consumer Accounts** *[b5/p38]* highly recommended to migrate consumer user accounts to managed user accounts

**Google Compute Engine Service Account Workflow** *[b1/p151]* Diagram of creation of GCP compute instance and the association of a custom service account

**Google Firebase: Acquisition** *[b4/p93]* Acquired in 2014 by Google: Google + Firebase, Divshot = New Firebase

**Google Firebase: Admin SDK Service Agent** *[b4/p122]* Upon activation, Firebase automatically creates the following SA and Project-level IAM roles: Google Managed SAs, Customer-Managed SA

**Google Firebase: Assessment Criteria** *[b4/p97]* As Firebase is an entire platform, assess services indepently

**Google Firebase: Authentication** *[b4/p112]* Few custom authentication schemes: Email / Pass, Phone, Anonymous

**Google Firebase: Authentication - Anonymous Provider** *[b4/p113]* App can authenticate a user without any user interaction

**Google Firebase: authentication - email / phone templates** *[b4/p113]* Critical components include email address verification, password reset flow, and ability to change user's email

**Google Firebase: Cloud Firestore Data Corruption** *[b4/p104]* Example Node.js that takes config from frontend app and uses it to interact with Firestore

**Google Firebase: Cloud Firestore Extraction** *[b4/p103]* Firestore Explorer example, retrieve the contents of the specified location

**Google Firebase: Combining GCP and Firebase Projects** *[b4/p121]* Concerning: user roles and permissions for your project will be shared

**Google Firebase: Compliance Concerns** *[b4/p125]* Most notably, Firebase does not allow the customer to define where its data is stored and processed

**Google Firebase: Database Defense** *[b4/p111]* Security rules to control authentication and authorization, monitor rule evaluation metrics

**Google Firebase: Database Insecure Rules Alert** *[b4/p107]* Firebase takes precautions to alert customers of insecure rules

**Google Firebase: Database Security Rules - Hunt the Bug** *[b4/p115]* Bad example: Authenticated users can view and corrupt data from other authenticated users

**Google Firebase: Database Security Rules - Improved** *[b4/p116]* Confirming session ID ensuring interaction matches the user's ID

**Google Firebase: Database Security Rules - Legacy Default** *[b4/p108]* Both Firebase DBs make broken auth too easy

**Google Firebase: Database Security Rules - Test Mode** *[b4/p105]* Example config for Test Mode - turn of all authentication and authorization for 30 days

**Google Firebase: Database Test Mode Alert** *[b4/p106]* Firebase takes precautions to alert customers of expiring access

**Google Firebase dump** all records in Realtime DB, grants anonymous users read access – only need Database name

**Google Firebase: Extensions** *[b4/p118]* Currently 14 extensions

**Google Firebase: History** *[b4/p91]* Founded in 2012, created DB which featured real-time synchronization and direct data access

**Google Firebase: HospitalGown Vulnerability** *[b4/p99]* Frontend app that might be secure, but it leverages a wide-open backend db that has not been configured to require proper authentication or authorization | read, write rules set to “true”

**Google Firebase: Hosting** *[b4/p117]* Used to easily deploy static websites

**Google Firebase: Incompatible Org Policy** *[b4/p124]* may be incompatible with iam.disableServiceAccountCreation

**Google Firebase: Privilege Escalation via Firebase Admin SDK Agent** *[b4/p123]* Create compute instance > config compute to run as the firebase admin SA > obtain token through IMDS > impersonate privileged SA in project

**Google Firebase: PUT, POST, PATCH, DELETE** *[b4/p102]* Example replace contents of a misconfigured database

**Google Firebase: Realtime Database Data Corruption** *[b4/p102]* Example replace contents of a misconfigured database

**Google Firebase: Realtime Database Extraction** *[b4/p101]* Example dump and parse all contents of a misconfigured database

**Google Firebase: Realtime Database Reconnaissance** *[b4/p109]* All DBs are located at a subdomain of firebaseio.com

**Google Firebase: Realtime Database vs Cloud Firestore** *[b4/p98]* Cloud firestore is the new, Realtime Database

will NOT be deprecated, but unlikely to get any major updates

**Google Firebase: Services** *[b4/p96]* Fully-fledged cloud platform with 18 services and counting

**Google Firebase: Summary** *[b4/p126]* Firebase is unique, services are insecure by default and easy to misconfigure - need to evaluate if it's worth to use

**Google Firebase: Why Firebase Matters** *[b4/p94]* Very popular, ease of use, appealing for prototypes

**Google Groups: GCP** *[b1/p141]* In GCP, grouping of users is done through Google Groups

# Hh

**HA VPN Gateway Example: GCP** *[b2/p143]* More advanced offering two external IP addresses and the ability to create two different tunnels

**HashiCorp Language (HCL)** *[b1/p28]* purpose is to define "resources" in code, used for Terraform

**https_put_response_hop_limit = 1** | prevent routing to IMDS beyond the requesting instance

**HSM: AWS CloudHSM** *[b3/p17]* single-tenant dedicated HSM, more expensive

**HSM: Azure Dedicated HSM** *[b3/p26]* Similar to AWS CloudHSM

**HSM: Google Cloud HSM** *[b3/p33]* Does NOT imply single-tenancy like AWS CloudHSM - only available for "selected customers"

# Ii

**IAM additional Statement Elements: AWS** *[b1/p87]* Sid, Principal, NotPrincipal, NotAction, NotResource, Condition

**IAM Administrative Assessment Criteria** *[b1/p85]* AWS CIS Benchmark 1.22 - Ensure IAM policies that allow full *.* admin privileges are not created

**IAM Better Policy Example: AWS** *[b1/p86]* better policies use action and resource restrictions

**IAM Instance Role Assessment Criteria: AWS** *[b1/p82]* AWS CIS Benchmark 1.19 requires all AWS virtual machines to use instance profile roles for managing credentials

**IAM Intro** *[b1/p42]* Limit access to managed services. Two entities: individuals and infra.

**IAM Key Terms: AWS** *[b1/p79]* Principal, Root user, IAM user, IAM Group, IAM Role, Instance Profile, Policy

**IAM Key Terms: GCP** *[b1/p124]* Member, Role, Policy

**IAM Policies: AWS** *[b1/p83]* write policies with as few permissions as necessary, defined with statements - contain Effect, Action, Resource

**IAM Policy Condition Requiring Private Access: AWS** *[b2/p102]* IAM Terraform, contain checks validating the origin VPC

**IAM Policy Summary** *[b1/p162]* CSP comparison of: Organization Policy, Principal Policy, Resource Policy, Conditional Policy, Default SA Policy

**IAM User - Creation with Terraform: AWS** *[b5/p12]* Terraform - similar to creating an IAM role

**IAM User - Terraform State with PGP Key: AWS** *[b5/p14]* Terraform module supporting providing a pgp_key argument, to avoid persistence

**IAM User - Terraform State: AWS** *[b5/p13]* Terraform will cache access key ID and secret access key in the unencrypted terraform.tfstate file

**IBM Cloud** *[b5/p98]* Next big 3 - Alibaba, IBM, Oracle; still niche players by Gartner

**Identities with Long-Lived Creds** *[b5/p9]* AWS IAM Users, Azure AD Service Principals, GCP Service Accounts

**Identity Key Terms: Azure** *[b1/p106]* Principal, Managed Identity, RBAC, Conditional Access | some terms are the same with CSPs, but others are drastically different

**Identity Namespace: GCP** *[b1/p149]* Swiss army knife for GKE, feature that keeps durable secrets out of your containers

**Identity Policy Example: AWS IAM** *[b1/p95]* attach directly to an IAM user, group, or role - defines actions principal can perform on resources

**IMDS Assessment Criteria: AWS** *[b1/p69]* Turn off endpoint, require tokens, set hop limit (TTL) - codify hardened config with Terraform

**IMDS v2 control** [b1/p69] prevent token extraction, use TTL

**IMDS Assessment Criteria: GCP** *[b1/p70]* turn off legacy endpoints vulnerable to SSRF, v0.1 and v1beta1

**Impersonation is a Tool for Lateral Movement** *[b4/p120]* Transitive Path: Impersonate SA > impersonate every SA in a project > has privileged role > profit

**Instance Metadata** *[b1/45]* IM elements

**Instance Metadata API Multicloud Summary** *[b1/p71]* Table comparison of metadata API security controls: SSRF Protection, Token Timeout, Token Scope, Requires REST API, Prevents Extraction

**Instance Metadata Service Example: Azure** *[b1/p47]* query Azure VM IMDS API, runs on 169.254.169.254

**Instance Metadata Server Example: GCP** *[b1/p48]* query GCE IMDS for network security groups, runs on 169.254.169.254

**Instance Metadata Service v1 Example: AWS** *[b1/p46]* query EC2 IMDS for network security groups

**Instance Profile Creds (IMDSv1): AWS** *[b1/p53]* temp access keys are valid for 6hrs

**Instance Profile Creds (IMDSv2): AWS** *[b1/p54]* Nov 19th, 2019 major upgrades to IMDSv2 in response to Cap 1 breach. Controls for Open WAFs, Open Reverse Proxies, SSRF, Credential Theft

**Instance Profile Workflow: AWS** *[b1/p81]* create role > launch instance with role > app retrieves role > app assumes role

**Internet Gateway: AWS** *[b2/p12]* Connect resources within a VPC to the internet

# Jj

**JSON Web Token (JWT) Function Request** *[b4/p24]*
**JWT AUD** *[b1/p57]* Audience - Resource Endpoint
**JWT verify authenticity** *[b5/p34]* use JWKS

# Kk

**Key Vault Access Policy Example: Azure** *[b3/p23]* Terraform - grant every possible permission for our primary key vault to the current principal

**Key Vault Deletion Window: Azure** *[b3/p25]* Azure Key Vault does NOT perform soft delete - you can enable **soft-deletion** and **purge protection**

**Key Vault Example: Azure** *[b3/p22]* Terraform - create key vault, generate RSA key and store it

**Key Vault Important Terms: Azure** *[b3/p21]* Secret, Vault, Vault Owner, Vault Consumer

**Key Vault Overview: Azure** *[b3/p19]* umbrella service for storing sensitive data - unlike KMS, can be used to store secrets and certificate mgmt | shortest retention period before permanent key deletion

**KMS and External Master Keys: AWS** *[b3/p16]* Customer can choose to generate key externally and import it to KMS

**KMS Audit Logging with CloudTrail: AWS** *[b3/p14]* Terraform - log all AWS API called to S3 via CloudTrail

**KMS Infrastructure: AWS** *[b3/p9]* Service interface to a highly secured system of HSMs, which host and protect customer master keys

**KMS Key Resource-Based IAM Policy Example: AWS** *[b3/p13]* Key Policy Example

**KMS Key Rotation Schedules: GCP** *[b3/p29]* **NOT** automatically rotated, can be manually rotated on-demand, customers can define rotation period, only supports SYMMETRIC keys

**KMS Key Terms: AWS** *[b3/p10]* CMK, Data Key (DK), Encryption Context, Envelope Encryption

**KMS Key Terms: GCP** *[b3/p28]* Key ring, Key, Key version, Primary key version

**KMS Overview 2: AWS** *[b3/p8]* Master key durability, Auditable, Safe and immediate master key deactivation

**KMS Overview: AWS** *[b3/p6]* HSM-secured master key creation and preservation, Automatic symmetric master key rotation, seamless integration with most AWS services

**KMS Overview: GCP** *[b3/p27]* Blend of AWS KMS and Azure Key Vault

**KMS Usage Example: AWS** *[b3/p12]* Terraform - create KMS Key for use with Secrets Mgr secret

**KMS Usage Example: GCP** *[b3/p30]* Terraform - create key hierarchy to encrypt secret string

# Ll

**Lambda - API Gateway Integration: AWS** *[b4/p16]* Serverless Framework YAML config to deploy AWS Lambda

**Lambda - Execution Role: AWS** *[b4/p17]* Serverless Framework YAML config to create custom AWS Lambda execution role

**Lambda Environment - API Gateway: AWS** *[b4/p15]* API gateway integration allows for additional security controls

**Lambda Environment - Default Network Config: AWS** *[b4/p13]* Not publicly accessible over HTTP, permissive egress traffic flow from function's network

**Lambda Environment - Environmental Variables: AWS** *[b4/p12]* contain the execution role's access keys, active for 12hrs

**Lambda Environment - Persistence: AWS** *[b4/p52]* Runtime environments are read-only except for the /tmp directory

**Lambda Environment - Source Code: AWS** *[b4/p11]* located in /var/task directory | files **handler.hjs, config.js** may contain insecure secrets

**Lambda Security Controls: AWS** *[b4/p14]* Hardening Lambda environment with several default config modifications

**Lambda VPC Config: AWS** *[b4/p19]* Serverless Framework YAML VPC config that moves Lambda function's execution environment into a customer managed VPC

**Lifecycle Config S3** *[b3/p77]* transitions/delete expires obj

**Link-local IP Address** *[b1/p46]* IP of IDMS service – 169.254.169.254

**Log Azure Key Vault Events to an Analytics Workspace** *[b3/p24]* Terraform for Azure Monitor = AWS CloudTrail

**Long-lived IAM Cred Mgmt: AWS** *[b5/p10]* programmatically reference creds through Parameter Store

**Long-Term IAM Cred Mgmt: AWS** *[b5/p10]* Many of AWS's best practices apply to all 3 CSPs

# Mm

**Managed Identity Credentials (IMDS): Azure** *[b1/p56]* Azure requires metadata request when retrieving JWT

**Managed Identity JWT: Azure** *[b1/p57]* Audience scope of a JWT is limited to a single Azure REST API

**Managed Identity Workflow for VMs: Azure AD** *[b1/p108]* off by default for new Azure VMs, App Services, and Functions

**Managed Identity: Azure** *[b1/p109]* Azure Managed Identity is comparable to AWS Instance Profile roles | **automatically** authenticates apps to other Azure services **without** needing to manage creds

**Managed Policy: Overly Permissive Example - AWS** *[b1/p100]* This is the same policy that allowed the Capital One breach

**Managed VPC Endpoints: AWS** *[b2/p99]* Interface endpoints: ENI powered by AWS PrivateLink, can be accessed from same subnet via IP. Vast majority of AWS-managed endpoints are interface endpoints, | Gateway endpoints: only S3 and Dynamo DB use

**Managed vs Inline Policies: AWS** *[b1/p99]* AWS managed, customer managed, Inline

**Mergers & Acquisitions** *[b1/p17]* Make multicloud inevitable, 2 main options: lean the new tech or migrate acquired assets (too expensive)

**Metadata-Flavor: Google** [b1/p48] prevents SSRF

**MFA Delete S3** *[b3/p77]* requires physical device or code

**Microsoft Graph** *[b5/p30]* id_token will contain data request by app, Graph API support many diff permissions

**Microsoft Identity Platform - Configuring a Web App** *[b5/p32]* Node.js app for OIDC

**Microsoft Identity Platform - ID Token Decoded 1** *[b5/p35]* JSON payload from ID token with the profile scope

**Microsoft Identity Platform - ID Token Decoded 2** *[b5/p36]* Continued from 1

**Microsoft Identity Platform - Permissions Requests** *[b5/p33]* Example of app requesting two relatively innocuous permissions

**Microsoft Identity Platform - Validating the Token** *[b5/p34]* decoding and validating JWTs alongside their library for interacting with a JWKS to validate a token against the Azure AD tenant

**Microsoft Identity Platform OIDC Flow** *[b5/p29]* OIDC diagram from Azure docs

**Misconfigured Virtual Machines Access on GCP** *[b2/p4]* GCP Compute Engine are unnecessarily accessible by default, can search on Shodan

**MITRE ATT&CK Cloud Matrix** *[b1/p21]* common techniques to attack Big 3, each cloud matrix covers Initial Access > Persistence > PrivEsc > Defense Evasion > Credential Access > Discovery > Collection > Exfil > Impact

**MITRE ATT&CK Cloud Services** *[b1/p23]* guides attack methodology for the Big 3 key cloud services, can easily identify Compute, Networking, IAM, Storage, Key Mgmt, Database

**Multicloud App Service Security Summary: Azure & GCP** *[b4/p87]* Comparing security of app service platforms: Shell Access, Default SA, Insecure Traffic, FTP Deployment

**Multicloud Architecture** *[b1/p26]* Use CI/CD to apply IaC (TF) to multicloud environments, AWS CodePipeline, Azure DevOps, GCP Continuous delivery

**Multicloud Benefits** *[b1/p14]* Use best services and unique benefits of each provider, avoid vendor lock-in, DR

**Multicloud Chart** *[b1/p13]* 86% respondents identified as multicloud, average is 3 CSPs

**Multicloud Data Exfiltration Summary** *[b3/p129]* Comparing the public data exfiltration options: Disk Snapshots, Database Snapshots, Signed URLs, Misc. Resources

**Multicloud Default Network Configuration** *[b2/p57]* Comparing the default network config and traffic flow options: Connected to the internet, Admin ports open, Ingress filtering, Egress filtering, Consistent controls

**Multicloud Default Network Settings** *[b2/p9]* Understand how open cloud networks are by default, then lock it down. This slide has the questions to ask.

**Multicloud Drawbacks** *[b1/p15]* Can't keep track of all services, managing access between providers - cred rotation, key mgmt, network peering

**Multicloud Flow Logging Summary** *[b2/p87]* Comparing network traffic flow logging options: Enabled by default, Minimum delay, Max retention period, CLI support, Log blocked ingress traffic

**Multicloud Integration Use-Cases** *[b5/p3]* Enables use of unique services not available in primary CSP, BC/DR

**Multicloud Key Management Service Summary** *[b3/p34]* Comparing the cloud-managed key mgmt options: Flexible access policy, Audit logging, Auto Key Rotation, Deletion schedule, Single-tenant option

**Multicloud Private Endpoint Summary** *[b2/p114]* Comparing private endpoint security options: Internal Service Routes, Custom service endpoints, Service access control, Endpoint policy, Principal restrictions

**Multicloud Security Assessment** *[b1/p19]* Understand configs, establish baselines and policies, identify any deviations

**Multicloud Serverless Environment Summary** *[b4/p58]* Comparing the Serverless platform's environment security: Root User, Warm Environment, Credential Timeout, Read-Only File System, Default Network

**Multicloud Serverless Platform Security Summary** *[b4/p60]* Comparing the Serverless platform security options: Default SA, Custom SA, HTTPS Access, VPC Integration

**Multicloud Storage Summary** *[b3/p107]* Comparing the Storage platform's security options: Block Public Access Policy, Access Logging, Default Encryption, Data Retention

# Nn

**NAT gateway resource: AWS** *[b2/p22]* Terraform example creating VPC resources

**NAT gateway: AWS** *[b2/p21]* NAT enables instances in private subnets to access the internet

**Network Access Control Lists (NACLs)** *[b2/p13]* Provide STATELESS traffic flow at the VPC SUBNET level

**Network Assessment Criteria: Azure** *[b2/p37]* Benchmark 6: 6.1-ensure RDP restricted from internet, 6.2-ensure SSH "", 6.3-ensure SQL DBs do not allow ingress 0.0.0.0/0

**Network Assessment Criteria: GCP** *[b2/p47]* Benchmark 3 Networking: 3.1 - ensure the default network does not exist in a project, 3.6 - ensure that SSH access is restricted from the internet, 3.7 - ensure RDP is restricted from internet

**Network Logging Assessment Criteria: AWS** *[b2/p65]* Benchmark 2.9: Ensure VPC flow logging is enabled on all VPCs

**Network Logging Assessment Criteria: Azure** *[b2/p74]* Benchmark 6.4 Network security group flow logs should be enabled, and the retention period is set to greater than or equal to 90 days

**Network Logging Assessment Criteria: GCP** *[b2/p81]* Benchmark 3.9: Ensure VPC flow logs is enabled for every subnet in VPC network

**Network Security Group Default Rules: Azure** *[b2/p33]*

**Network Security Group: Azure** *[b2/p30]*

**Network Service Scanning** *[b2/p3]* How to automatically detect open ports on servers in each CSP IP space

**Network_watcher_name, resource_group_name** [b2/p76] logging config for NSG

**node.js retrieve credentials** [b5/p19] getSecret function

**NSG** | NSG & Azure Firewall are both **STATEFUL**

**NSG Flow Log Config 2: Azure** *[b2/p77]* Enabling flow logs for NSG with Traffic Analytics

**NSG Flow Log Config: Azure** *[b2/p76]* Enabling Azure flow logs using Terraform is more verbose than with AWS

**NSG Flow Log Example: Azure** *[b2/p79]* Azure's records are aggregates, not individual flow logs - command line call to retrieve flow log data from Azure log analytics

**NSG Flow Logs Querying with Traffic Analytics: Azure** *[b2/p78]* Similar to CloudWatch, Azure Log Analytics Workspaces used to query many types of logs, uses query language called Microsoft Kusto

**NSG Flow Logs: Azure** *[b2/p75]* Stores metadata about the traffic within a NSG to an Azure storage account

# Oo

**OAuth 2.0, OpenID Connect, and SAML** *[b5/p24]* OAuth to authorize, OpenID adds authentication layer, SAML is XML alternative to OIDC

**Object Lock S3** *[b3/p77]* stores using WORM model

**Object Versioning S3** *[b3/p77]* multiple versions of each obj

**Outage 2017 AWS S3** *[b5/p4]* Four hour outage

**Outage 2018 Azure** *[b5/p6]* Lightning struck

**Outage 2020 AWS Kinesis Data Streams** *[b5/p5]* Capacity changes, Orgs can do BC/DR by using multiple regions

**Outage 2020 Google Cloud** *[b5/p7]* 1 hr outage due to authentication system outage

**Open-Source - GCP CIS 1.1.0 InSpec Profile** *[b5/p78]* GCP CIS benchmark scan against a given project id

**Open-Source - GCP Config Validator** *[b5/p79]* contains example templates and constraints for many of the misconfigs

**Open-Source - GCP Security Response Automation** *[b5/p88]* Security Response Automation project architecture for Security Command Center notifications

**Oracle Cloud** *[b5/p98]* Next big 3 - Alibaba, IBM, Oracle; still niche players by Gartner

**Org Policy Constraints: GCP** *[b1/p134]* Config of restrictions to IAM Policy, closest thing GCP has to blocking inheritance of permissions | Org > Folders > Projects > Resources

**Organization - Disable Public Blog Access: Azure** *[b3/p84]* Policy def that prevents the creation of new Azure Storage accounts

**Organization Conditional Access: Azure** *[b1/p122]* For enforcing organization wide access control policies, Azure AD Premium feature only

# Pp

**Packet Mirroring: GCP** *[b2/p86]* Similar to AWS Traffic Mirroring and Azure TAP.

**Pacu: AWS Exploitation Framework** *[b1/p78]* open-source framework by Rhino Security Labs - abuse IAM permission misconfigs

**Persistence with Serverless** *[b4/p48]* Runtimes are ephemeral by nature, runtimes are NOT destroyed and

recreated on each invocation, use storage services and DBs for permanent persistence

**Persistence with Serverless - Cleanup** *[b4/p51]* Need this line of code to prevent local file inclusion or command injection vulnerability

**Persistence with Serverless - Hunt the Bug** *[b4/p50]* Node.js - leverage open-source library jimp to transform the uploaded image

**Persistence with Serverless - Network-Protected Storage** *[b4/p57]* All 3 CSPs can be configured to integrate with the private network

**Persistence with Serverless - Proper Long-Term Storage** *[b4/p56]* DO NOT store data on a function's filesystem if you need to access it in the future, DO securely store data externally

**Point-to-site Client Config: Azure** *[b2/p136]* Downloading and configuring the endpoint device

**Policy Condition: Request Attribute Expressions: GCP** *[b1/p137]* Use details of request to create powerful condition expressions, very powerful in conjunction with Google Zero Trust offering - Identity-Aware Proxy (IAP)

**Policy Condition: Resource Attribute Expressions: GCP** *[b1/p138]* Compare properties of requested resource or of the request itself

**Policy Conditions: GCP** *[b1/p136]* Constrain access to a subset of GCP Resources with Policy Conditions, uses Common Expression Language (CEL)

**Policy Evaluation Logic: AWS IAM** *[b1/p92]* Explicit Deny > Explicit Allow > default deny unless there is explicit allow | **Priority** = Deny > SCPs > Resource > Permissions > Session > Identity

**Policy Types: AWS IAM** *[b1/p90]* SCP, Resource-based, Identity-based, Session, Permissions Boundary, ACLs

**Primitive Roles as Anti-Patterns: GCP** *[b1/p130]* Owner Role, Editor Role, Viewer Role

**Principal Example: AWS IAM** *[b1/p88]* only allow request if originated from trusted role or user

**Private Access IAM Policy Validation: AWS** *[b2/p103]* validate IAM condition is working, private endpoint

**Private Endpoint DNS: Azure** *[b2/p109]* Update internal DNS for Vault to reference private endpoint in terraform

**Private Endpoint Example: Azure** *[b2/p108]* Azure Private Endpoint for Key Vault in terraform

**Private Endpoint VNet Config: Azure** *[b2/p110]* Terraform to associate PL service, enable network policies for private link service

**pgp_key** *[b5/p14]* encrypt tf config from cached creds

**Private Endpoint: Azure** *[b2/p107]* allows VNet resources to connect privately to a Private Link service

**Private Link Services: Azure** *[b2/p106]* Examples of services that azure hosts PL endpoints for

**Private Link: Azure** *[b2/p105]* Has a space between, similar to AWS PL - allows access to Azure PaaS or customer-owned services in your VNet

**Private Subnet Resource: AWS** *[b2/p23]* Terraform solution creating private subnet and route to NAT gateway

**PrivateLink - Custom Services: AWS** *[b2/p104]* Allows service providers to publish a custom endpoint service for service consumers

**Privilege Escalation: Azure** *[b1/p104]* Resources for Azure PrivEsc: DerbyCon 9, PowerZure

**Prowler** *[b3/p112]* Added to Security Hub, by Toni Blyx

**Public Cloud Service Default Config** *[b2/p95]* By default, VPC resources communicate with cloud services over the internet, multitenant

**Public Data Exfiltration Paths: AWS Prowler** *[b3/p112]* AWS Prowler contains several audit checks for additional public resources

**Purge Protection** *[b3/p25]* Azure Key Vault does NOT perform soft delete - you can enable soft-deletion and purge protection

# Qq

**Query Azure Traffic Analytics** [b2/p75] az log-analytics

# Rr

**RBAC Custom Roles: Azure AD** *[b1/p120]* RBAC def allowing read access to a single Azure storage container

**RBAC Limitations: Azure AD** *[b1/p121]* Not the only authorization mechanism in Azure, can use Shared Access Signatures or SAS tokens

**RDS Disk Level Encryption** *[b3/p46]* if no KMS specified, will use **default** AWS-managed CMK

**RDS - Sharing Database Snapshots: AWS** *[b3/p114]* Sharing the database backups should be monitored closely to ensure attackers are not providing a path to exfiltrate data

**Record-level Encryption with AWS KMS** *[b3/p47]* Node.js - encrypt and serialize plaintext secret using AWS KMS SDK

**Record-level Encryption with Azure Key Vault** *[b3/p55]* Node.js Key Vault encrypt/decrypt data

**Record-level Encryption with Google Cloud KMS** *[b3/p58]* Node.js - use cloud KMS key to encrypt/decrypt data

**Resource Hierarchy and Inheritance: GCP** *[b1/p126]* Resources organized hierarchically, each resource has exactly one parent.

**Resource Manager** | deployment & mgmt. service for Azure resources, used to enable Managed Identities

**Resource Policy Example: AWS IAM** *[b1/p94]* attach directly to resource - S3 bucket, KMS key, SQS queue, etc

**Resource Policy S3** *[b3/p111]*

**Resource Provider Operations: Azure** *[b1/p115]* defines the operations available to use in a role definition, hundreds or operation strings

**Revenue Trends** *[b1/p6]* Growing, big $$, Microsoft does not disclose figures, Google sometimes discloses

**Role Assignment Scopes: Azure** *[b1/p119]* Assignments include All, Management Group, Subscription, Resource Group, Resource

**Role Definition: Azure** *[b1/p113]* collection of permissions that define the allowed/disallowed operations using properties: Actions, NotActions, DataActions, NotDataActions, AssignableScopes

**Role-Based Access Control (RBAC): Azure AD** *[b1/p112]* Azure relies on RBAC to manage access to resources, contains principal, role definition, scope

**Roles - Organization Admin: GCP** *[b1/p128]* Full access to Resources in the Organization, most powerful is the G Suite Super Admin

**RSA key-pair: GCP** *[b1/p143]* used for SA authentication

# Ss

**S3 - CloudTrail Object Logging** *[b3/p73]* Need to config CloudTrail to include events from the relevant S3 buckets

**S3 - Sharing Private Objects: AWS** *[b3/p115]* shared externally by creating a presigned URL

**S3 Assessment Criteria** *[b3/p67]* benchmark: storage, IAM, and data governance

**S3 Bucket Hardening - Access Logging** *[b3/p74]* Terraform - Provides detailed audit logs of all requests made to the bucket

**S3 Bucket Hardening - Block Public Access** *[b3/p72]* Terraform - Bucket level public access block policy

**S3 Bucket Hardening - CMK Encryption** *[b3/p75]* Terraform - apply KMS encryption for all S3 objects

**S3 Bucket Hardening - Data Retention** *[b3/p77]* Must be configured to have object versioning, MFA delete, object lock, lifecycle config

**S3 Bucket Hardening - Lifecycle Config** *[b3/p79]* Terraform - lifecycle config rule

**S3 Bucket Hardening - Secure Transport** *[b3/p76]* Bucket policy config denying all principals and all actions made to bucket without secure transport

**S3 Bucket Hardening - Versioning and Object Locking** *[b3/p78]* Terraform - versioning and object lock config

**S3 Bucket Policy for CloudTraill: AWS** *[b3/p15]* Resource-based policy that we can attach to the bucket to grant CloudTrail permissions

**S3 Control Service - Account Level Config** *[b3/p70]* Config for S3 Control Service to block all public access

**S3 Control Service - Account Level Example** *[b3/p71]* Querying S3 control service for acc level public access block settings

**S3 Control Service - Account Level Settings** *[b3/p69]* Query control service to check account level public access block settings

**S3 Presigned URL - CloudTrail Detection: AWS** *[b3/p117]* Use CloudTrail and CloudWatch to detect usage of presigned S3 URLs

**SaltStack: Cloud Network Security Gone Wrong** *[b2/p5]* SS used to configure cloud infra, vuln allowed for Remote Code Execution (RCE) as root, 6k public salt master exposed, servers on private network were not exposed

**Scoped KMS Role Binding: GCP** *[b3/p31]* Terraform - create custom IAM role that grants minimum number of permissions to allow principal to decrypt string with KMS key

**SCP: AWS IAM** *[b1/p93]* Service Control Policy: apply centralized policies for one-to-many sub-accs

**Secure Access Key Storage** *[b5/p18]* Terraform - store as secrets

**Security Group Egress Rule: AWS** *[b2/p25]* Terraform solution creating a security group, egress

**Security Group Ingress Rule: AWS** *[b2/p24]* Terraform solution creating a security group, ingress

**Security Groups: AWS** *[b2/p15]* At the INSTANCE level, STATEFUL, control ingress and egress traffic flow to EC2 network interface

**Security Group** | only **ALLOW**, most permissive rule wins

**Send additional Cloud KMS Logs to Google Cloud Logging** *[b3/p32]* Terraform - turns logging on for all DATA_READ events in Cloud KMS

**Serial Console Config: Azure** *[b2/p129]* Requires boot diagnostics, local user, grant VM contributor to both

**Serial Console Example: Azure** *[b2/p130]* Shows private connection to an Azure VM using Azure Serial Console

**Serial Console: Azure** *[b2/p128]* Azure's session manager

**Serverless Framework - Overview** *[b4/p6]* tool (The Serverless Framework), IaC like Terraform, uses YAML

**Serverless Prey - Overview** *[b4/p8]*

**Server-side Request Forgery** *[b1/50]* SSRF overview

**SSRF** | **disable-legacy-endpoints = "TRUE"** to reduce change of SSRF on GCP

**Service Account Credentials (v0.1 and v1beta1): GCP** *[b1/p58]* vulnerable endpoints are deprecated and will be shut down, but no date from Google

**Service Account Credentials (v1): GCP Computer Engine** *[b1/p60]* v1.0 metadata service requires a custom header to submit a request

**Service Account Impersonation Abuse: GCP** *[b1/p147]* Three categories for GCP member to use to operate a Service Account: Impersonation, ActAc, Create Keys

**Service Account with Key - Creation with Terraform: GCP** *[b5/p17]* Terraform resource we use to create SA for the VM in GCE

**Service Accounts - GCP Super Hwy for Lateral Movement** *[b4/p119]* Hopscotch to Privilege via SAs and their role bindings

**Service Accounts as Both Identities and Resources: GCP** *[b1/p146]* Service Accounts in GCP are both an identify and a resource

**Service Network Address: Azure** *[b2/p111]* Terraform to allow endpoint network access config

**Session Manager** | access granted using IAM policies, use ssm:StartSession action

**Session Manager Config: AWS** *[b2/p122]* Session Mgr needs running EC2 with SSM Agent v2.3.68 or greater

**Session Manager Example: AWS** *[b2/p123]* Connecting to private instance using AWS web console session mgr

**Session Policy Example: AWS** *[b1/p98]* Session policy file hardening wide-open S3 inherited access policy

**Session Policy Overview: AWS IAM** *[b1/p96]* used when assuming the permissions of a predefined role in SSO, federated identity, and web identity auth flows

**Session Token Creation: AWS IAM** *[b1/p97]* Select base role with identity-based policy > add resource based policy > add session based policy

**Shadow Cloud Accounts** *[b1/p18]* cloud acc not managed or sanctioned by enterprise, unregulated, bypasses procurement - reintegration is costly

**Sharing Disk Snapshots: GCE** *[b3/p124]* Google Compute Engine (GCE) VM disk snapshot in the web console

**Shared Access Signature (Storage)** | expiration window is required, need --duration-in-seconds

**Shodan** *[b2/p3]* enumeration, service discovery

**SMB 2.1, 3.0** *[b3/p90]* secure transfer

**Soft deletion** *[b3/p25]* Azure Key Vault Deletion Window

**SQL Encryption (MSSQL)** *[b3/p51]* uses TDE

**SR Presigned URL Considerations: AWS** *[b3/p116]* Monitor CloudTrail for calls to the S3 presign, consider adding S3 bucket policies that restrict object access to trusted IP Addresses

**SSH from the Browser - Firewall Rules: GCP** *[b2/p139]* Does not automatically allow internal GCP traffic

**SSH from the Browser: GCP** *[b2/p137]* Similar to AWS SSM Session Mgr

**SSO: OAuth 2.0, OIDC, SAML** *[b5/p24]*

**SSO - Accessing AWS Accounts in Org: AWS** *[b5/p27]* Permissions created via Control Tower, service used to manage AWS Orgs. SSO support 3rd party services

**SSO: AWS** *[b5/p26]* SSO across multiple accounts within single AWS Org OU, supports SAML 2.0

**Standardization** *[b1/p16]* CCoE or Cloud Gov, too much is bad: business needs come first, do not slow down devs

**Stolen Credential Pivoting** *[b1/p61]* Different in each CSP, AWS easy, Azure & GCP difficult

**Stolen Creds: AWS** *[b1/p62]* to work: set access key ID, secret access key, session token environment variables

**Storage - Azure Monitor Diagnostic Setting** *[b3/p86]* Terraform - create an Azure Monitoring Diagnostic Setting

**Storage - Bucket Access Control IAM***[b3/p98]* Two diff bucket access control options: Fine-grained Access Control, Uniform Access Control

**Storage - Container Access Levels: Azure** *[b3/p82]* Support Container, Blob, Private | Terraform for containers

**Storage - Data Retention/Audit: Azure** *[b3/p91]* Must enable several key features to ensure Blobs are secure: Immutable Policies and Lifecycle Requirements

**Storage - Detecting Access to Sensitive Files: GCP** *[b3/p128]* Monitor any API call to download a file that is sensitive and rarely accessed

**Storage - Detecting SAS Use with Log Analytics: Azure** *[b3/p123]* query Azure Log Analytics for StorageBlobLogs

**Storage - Disable Public Blob Access: Azure** *[b3/p83]* Terraform - set Allow Blob Public Access to false

**Storage - SAS with Terraform: Azure** *[b5/p16]* Shared Access Signature (SAS) can be used to access data in an Azure Storage account

**Storage – SAS Shared Access Signature Considerations: Azure** *[b3/p121]* SAS Benchmarks

**Storage - SAS Sharing Private Storage Objects: Azure** *[b3/p119]* Users can share containers or Blob objects by generating Shared Access Signatures (SAS)

**Storage - Logging data Actions: Azure** *[b3/p85]* Need to write custom scripts or manually assess Storage accounts for audit logging, no Terraform support

**Storage - Public Access Control Example: GCP** *[b3/p96]* Cloud Storage bucket with allUsers member in Storage Object Viewer

**Storage - Public Binding Example: GCP** *[b3/p97]* Terraform - IAM binding granting allUsers the roles/storage.objectViewer role

**Storage - Public Bucket Detection: GCP** *[b3/p99]* Provides several native controls to identify and potentially remediate publicly accessible buckets

**Storage - URL Signing Considerations: GCP** *[b3/p127]* Disable service account key creation, restrict access to projects inside perimeter or access level expectations, use gsutil or SDK to view cloud audit logs

**Storage - URL Signing: GCP** *[b3/p126]* gsutil command generating a signed URL for an object stored in a private bucket

**Storage Assessment Criteria: Azure** *[b3/p80]* Benchmark CIS 3 - Storage Accounts, IAM, Governance

**Storage Assessment Criteria: GCP** *[b3/p94]* Benchmark CIS 5 - Storage, IAM, Data Governance

**Storage Hardening - Customer Managed Encryption: Azure** *[b3/p87]* Terraform - config for customer managed encryption key | Supports Infra Level Enc - Msoft Managed Keys, Service Level Enc - Customer provided keys, Service Level Enc - Customer managed keys

**Storage Hardening - Lifecycle Mgmt: Azure** *[b3/p93]* Blob object lifecycle rule moving audit data between storage tiers and eventually deleting blobs after retention period has been met

**Storage Hardening - Secure Transfer Required: Azure** *[b3/p90]* Terraform - enable Secure transfer required setting

**Storage Hardening: Immutable Container Policy: Azure** *[b3/p92]* No Terraform support, need to write custom scripts or manually assess Storage accounts for data retention policies

**System Assigned Managed Identity: Azure** *[b1/p110]* A cloud-managed identity solution for resources such as Azure VM, Azure App Service, and Azure Functions - is ephemeral

**Systems Manager (SSM) Session Manager: AWS** *[b2/p121]* SSM contains Session Mgr - allows IAM users to access EC2 instances **without** bastion hosts, remote desktop gateways, or SSH keys

# Tt

**T1046 MITRE ATT&CK** *[b2/p3]* How to automatically detect open ports on servers in each CSP IP space

**T1496 MITRE ATT&CK** *[b1/p77]* consuming victim's cloud resources to solve resource intensive problems

**T1522 MITRE ATT&CK** *[b1/p45]* service acc creds and config data stored in Instance Metadata Service (IMDS)

**T1526 MITRE ATT&CK** *[b2/p94]* enumerating the cloud services accessed by a system after gaining access (used by a CSC)

**T1530 MITRE ATT&CK** *[b3/p65]* improperly secured cloud storage object

**Terraform Overview** *[b1/p27]* Vendor agnostic, open-source IaC, config files written in HCL

**Timeout value cannot exceed 30** | lambda functions

**TLS** | only x < version **1.2**, **1.3**+ are secure

**TLS - Google Cloud SQL Require Transport Security** *[b3/p59]* Does not require TLS by default, config example through Terraform

**TLS - Using Client Certificate for Google Cloud SQL** *[b3/p60]* Terraform - require that client provide a client certificate for authentication

**TLS for Azure DB - Require Transport Security** *[b3/p56]* Terraform - require TLS for Azure DB MySQL

**TLS for MySQL, Aurora, and MariaDB - AWS RDS** *[b3/p49]* Query to require TLS for MySQL compatible engines

**TLS for PGSQL and MSSQL - AWS RDS** *[b3/p48]* DB Parameter group rejecting unencrypted connections

**Traffic Mirroring: AWS** *[b2/p72]*

**Types of Roles: GCP** *[b1/p129]* Primitive, Predefines, and Custom Roles

**Types of Service Accounts: GCP** *[b1/p143]* User managed service accounts vs Google managed service accounts

# Uu

**User Assigned Managed Identity: Azure** *[b1/p111]* Customer managed standalone identity that can be assigned to multiple service instances

**Using Access Keys** *[b5/p19]* Only useful if utilized by client, shows Node.js app using creds to authenticate SDKs for each CSP

# Vv

**Virtual Machine Overview - SEC510** *[b1/p24]* contains tools for managing multicloud: GitLab, Terraform, AWS CLI, Azure CLI, GCP CLI

**Virtual Machine Service Accounts** *[b1/p43]* AWS EC2, Azure VM, GCE, execute with predefined permissions

**Virtual Machines - Sharing Disk Snapshots: Azure** *[b3/p118]* Azure console view, Unlike AWS, Azure does not have a feature for permanently sharing the disk publicly

**Virtual Network (VPN) Gateway: Azure** *[b2/p131]* Provides VPN - site-to-site and point-to-site

**Virtual Network Terminal Access Point (TAP): Azure** *[b2/p80]* Similar to AWS Traffic Mirroring

**Virtual Private Cloud - VPC: AWS** *[b2/p10]* Provides dedicated network inside AWS account

**Virtual Private Cloud - VPC: GCP** *[b2/p42]* Similar to AWS VPC - main difference is GCP VCP is global resource, nots coped to a region. GCP network is powered by virtual networking stack called Andromeda

**VNet - Virtual Networks: Azure** *[b2/p26]* Provides building block resources for enabling private networking | **DDoS protection** applied by default

**VNet Configuration: Azure** *[b2/p27]* Customers responsible for VNet config settings: region, address space (IPv4, IPv6), subnets, security options

**VNet Default VM Confirmation Warning** *[b2/p35]* Azure provides additional warning before creating the resource

**VNet Network Security Group Default Rules: Azure** *[b2/p33]* Allow inbound from only VNet and Azure LB, allow all outbound to VNet and Inter - Rules: AllowVnetInBound, AllowAzureLoadBalancerInBound, DenyAllInBound, AllowVnetOutBound, AllowInternetOutBound, DenyAllOutBound

**VNet Network Security Group: Azure** *[b2/p30]* Contain collection of stateful inbound and outbound rules for a subnet or network interface

**VNet Security Group EgressRule** *[b2/p39]* Terraform egress RDP access

**VNet Security Group Ingress Rule** *[b2/p38]* Terraform restricting SSH access to an admin IP

**VNet Service Tags: Azure** *[b2/p32]* Tags represent a group of Azure managed IPs for NSGs or Azure Firewalls: VirtualNetwork, Internet, AzureLoadBalancer, AzureCloud

**VNet Subnet Configuration: Azure** *[b2/p29]* segmentation for VNet: address range, route table, NAT gateway, network security group, service endpoints

**VNet Virtal Machine Network Access: Azure** *[b2/p34]* Like AWS, VMs created in UI default to open admin acces: Linux auto SSH, Win auto RDP

**VNet Virtual Machine Default Network Security Group** *[b2/p36]* Azure uses 300 for the priority rule, provides one final warning if using default config

**VNet: Application Security Groups** *[b2/p40]* ASGs allows csc to tag VMs and reference the tag in a security rule

**VPC Endpoint Interface: AWS** *[b2/p100]* Creating VPC endpoint for secret mgr using Terraform

**VPC Endpoints: AWS** *[b2/p98]* Customer creates endpoint in their VPC to an AWS-managed service (S3, secrets mgr, etc).

**VPC Flow Log Config: AWS** *[b2/p67]* Enabling flow logs using Terraform

**VPC Flow Log Config: GCP** *[b2/p82]* Easiest out of the CSPs to set up with Terraform | sampling rate is key

**VPC Flow Log Example: AWS** *[b2/p69]* Command line call to retrieve flow log data from a log group

**VPC Flow Log Example: GCP** *[b2/p83]* Example of JSON, all flow logs are for accepted traffic

**VPC Flow Log Example: GCP 2** *[b2/p84]* Example 2

**VPC Flow Log Querying in CloudWatch Insights: AWS** *[b2/p71]* Analyze & visualize flow log data, run in the AWS console or CLI

**VPC Flow Logs Querying with Google Cloud Logging: GCP** *[b2/p85]* Analyze and visualize flow log data, run in console or CLI

**VPC Flow Logs: AWS** *[b2/p66]* Allow traffic METADATA to be captured within a cloud network

**VPC Peering** *[b2/p10]* Allow VPCs in diff regions to communicate with each other

**VPC Service Controls: GCP** *[b2/p113]* Creates perimeter for cloud managed services to limit access, REQUIRES additional paid subscriptions | prevent access from unauthorized networks with **stolen creds**

**VPC Traffic Mirroring: AWS** *[b2/p72]* Allows customers to copy traffic from EC2's ENI to another EC2 or external security and monitoring appliances

**VPN Gateway Example 2: Azure** *[b2/p134]* Create an Azure Virtual Network Gateway in Terraform - creating the configuration

**VPN Gateway Example: Azure** *[b2/p133]* Create an Azure Virtual Network Gateway in Terraform

**VPN Gateway Point-to-Site Config: Azure** *[b2/p132]* Config for Azure **point-to-site** connection

**VPN: AWS** *[b2/p124]* Managed service VPN, site-to-site VPN, client VPN, VPN cloudhub, Customer managed

**VPN Authorization Rule** *[b2/p127]*

**VPN Gateway Client/Point-to-site options** *[b2/p144]* GCP VPN does not support point-to-site connections

**Vulnerable to SSRF** *[b1/58]* GCP v0.1 & v1beta1

# Ww

**Web Console** | displays public bucket warning to use when bucket has **allUsers** member in Storage Object Viewer role

# Xx

# Yy

# Zz & #

**zz AWS Summary** *[b5/p95]* IAM, Network Controls, Encryption

**zz Azure Summary** *[b5/p96]* Questions Azure expert must be ready to answer, last paragraph

**zz GCP Summary** *[b5/p97]* BAD - Editor role assigned to default SAs, default firewall rules allow SSH and RDP

**0.0.0.0** | starting and ending from 0.0.0.0 = opening DB to traffic within Azure internal network