sonar RULES

Products

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C#
- CSS
- Flex
- Go
- HTML
- Java
- JavaScript
- Kotlin
- Objective C
- PHP
- PL/I
- PL/SQL
- Python
- RPG
- Ruby
- Scala
- Swift
- Terraform
- Text
- TypeScript
- T-SQL
- VB.NET
- VB6
- XML

Terraform static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your TERRAFORM code

All rules 50 Vulnerability 5 Security Hotspot 43 Code Smell 2

Tags

Search by name...

Security Hotspot

Using unencrypted EFS file systems is security-sensitive

Security Hotspot

Using unencrypted SQS queues is security-sensitive

Security Hotspot

Using unencrypted SNS topics is security-sensitive

Security Hotspot

Using unencrypted SageMaker notebook instances is security-sensitive

Security Hotspot

Using unencrypted Elasticsearch domains is security-sensitive

Security Hotspot

Using unencrypted RDS databases is security-sensitive

Security Hotspot

Using unencrypted EBS volumes is security-sensitive

Security Hotspot

Disabling logging is security-sensitive

Security Hotspot

Administration services access should be restricted to specific IP addresses

Vulnerability

Unversioned Google Cloud Storage buckets are security-sensitive

Security Hotspot

Disabling S3 bucket MFA delete is security-sensitive

Disabling logging is security-sensitive

Analyze your code

Security Hotspot Major aws gcp cwe owasp

Disabling logging of this component can lead to missing traceability in case of a security incident.

Logging allows operational and security teams to get detailed and real-time feedback on an information system's events. The logging coverage enables them to quickly react to events, ranging from the most benign bugs to the most impactful security incidents, such as intrusions.

Apart from security detection, logging capabilities also directly influence future digital forensic analyses. For example, detailed logging will allow investigators to establish a timeline of the actions perpetrated by an attacker.

Ask Yourself Whether

- This component is essential for the information system infrastructure.
- This component is essential for mission-critical functions.
- Compliance policies require this component to be monitored.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

Enable the logging capabilities of this component.

Sensitive Code Example

For Amazon S3 access requests:

```
resource "aws_s3_bucket" "myncompliantbucket" { # Sen
  bucket = "myncompliantbucketname"
}
```

For Amazon API Gateway stages:

```
resource "aws_api_gateway_stage" "api-v1" { # Sensitive
  deployment_id = aws_api_gateway_deployment.example.id
  rest_api_id   = aws_api_gateway_rest_api.example.id
  stage_name    = "v1-prod-api"
  xray_tracing_enabled = false # Sensitive
}
```

For Amazon Neptune clusters:

```
resource "aws_neptune_cluster" "cluster" {
  enable_cloudwatch_logs_exports = []  # Sensitive
}
```

For Amazon MSK broker logs:

Security Hotspot

Disabling versioning of S3 buckets is security-sensitive

Security Hotspot

Disabling server-side encryption of S3 buckets is security-sensitive

Security Hotspot

AWS tag keys should comply with a naming convention

Code Smell

Terraform parsing failure

Code Smell

```
resource "aws_msk_cluster" "sensitive_msk" {
  cluster_name = "sensitive_msk"
  logging_info {
    broker_logs { # Sensitive
      firehose {
        enabled = false
      }
      s3 {
        enabled = false
      }
    }
  }
}
```

For Amazon MQ:

```
resource "aws_mq_broker" "broker" {
  logs {  # Sensitive
    audit = false
    general = false
  }
}
```

For Amazon DocumentDB:

```
resource "aws_docdb_cluster" "docdb_omitting_logs" { #
  cluster_identifier = "DB Cluster Without Logs"
}
```

For Amazon Redshift:

```
resource "aws_redshift_cluster" "cluster" {
  cluster_identifier = "redshift-cluster"

  logging {
    enable = false # Sensitive
  }
}
```

For Amazon Global Accelerator:

```
resource "aws_globalaccelerator_accelerator" "accelerat
  attributes {
    flow_logs_enabled   = false  # Sensitive
    flow_logs_s3_bucket = "example-bucket"
    flow_logs_s3_prefix = "flow-logs/"
  }
}
```

For Amazon OpenSearch service, or Amazon Elasticsearch service:

```
resource "aws_elasticsearch_domain" "domain" {
  log_publishing_options {
    cloudwatch_log_group_arn = "arn:aws:logs:us-east-1:
    log_type = "AUDIT_LOGS"
    enabled = false # Sensitive
  }
}
```

For Amazon CloudFront distributions:

```
resource "aws_cloudfront_distribution" "cloudfront_dist
  default_root_object = "index.html"
}
```

For both Amazon Classic Load Balancing and Application Load Balancing:

```
resource "aws_lb" "load_balancer" {
  access_logs {
    enabled = false # Sensitive
    bucket = "mycompliantbucket"
    bucket_prefix = "log/lb-"
  }
}
```

For GCP Cloud Storage service:

```
resource "google_storage_bucket" "example" { # Sensitiv
  name     = "example"
  location = "US"
}
```

For GCP Region Backend Service:

```
resource "google_compute_region_backend_service" "examp
  name                            = "example"
  region                          = "us-central1"
  health_checks                   = [google_compute_reg
  connection_draining_timeout_sec = 10
  session_affinity                = "CLIENT_IP"
  load_balancing_scheme           = "EXTERNAL"
  protocol                        = "HTTPS"
}
```

For GCP VPC Subnetwork:

```
resource "google_compute_subnetwork" "example" { # Sens
  name          = "example"
  ip_cidr_range = "10.2.0.0/16"
  region        = "us-central1"
  network       = google_compute_network.custom-test.id
}
```

For GCP SQL Database Instance:

```
resource "google_sql_database_instance" "example" {
  name             = "example"
  database_version = "POSTGRES_11"
  region           = "us-central1"

  settings { # Sensitive
    tier = "db-f1-micro"
    ip_configuration {
      require_ssl  = true
      ipv4_enabled = true
    }
  }
}
```

For GCP Kubernetes Engine (GKE) cluster:

```
resource "google_container_cluster" "example" {
  name               = "example"
  location           = "us-central1-a"
  initial_node_count = 3
  logging_service    = "none" # Sensitive
}
```

Compliant Solution

For Amazon S3 access requests:

```
resource "aws_s3_bucket" "myloggingbucket" {
  bucket = "myloggingbucketname"
  acl    = "log-delivery-write"
}

resource "aws_s3_bucket" "mycompliantbucket" {
  bucket = "mycompliantbucketname"

  logging {
      target_bucket = "myloggingbucketname"
```

```
    target_prefix = "log/mycompliantbucket"
  }
}
```

For Amazon API Gateway stages:

```
resource "aws_api_gateway_stage" "api-v1" {
  deployment_id = aws_api_gateway_deployment.example.id
  rest_api_id   = aws_api_gateway_rest_api.example.id
  stage_name    = "v1-prod-api"
  xray_tracing_enabled = true
  access_log_settings {
    destination_arn = "arn:aws:logs:eu-west-1:123456789
    format = "..."
  }
}
```

For Amazon Neptune clusters:

```
resource "aws_neptune_cluster" "cluster" {
  enable_cloudwatch_logs_exports = ["audit"]
}
```

For Amazon MSK broker logs:

```
resource "aws_msk_cluster" "sensitive_msk" {
  cluster_name = "sensitive_msk"
  logging_info {
    broker_logs {
      firehose {
        enabled = false
      }
      s3 {
        enabled = true
        bucket  = "myloggingbucketname"
        prefix  = "log/msk-"
      }
    }
  }
}
```

For Amazon MQ enable `audit` or `general`:

```
resource "aws_mq_broker" "broker" {
  logs {
    audit = true
    general = true
  }
}
```

For Amazon DocumentDB:

```
resource "aws_docdb_cluster" "docdb_omitting_logs" {
  cluster_identifier = "DB Cluster With Logs"
  enabled_cloudwatch_logs_exports = ["audit"]
}
```

For Amazon Redshift:

```
resource "aws_redshift_cluster" "cluster" {
  cluster_identifier = "compliant-redshift-cluster"
  logging {
    enable          = true
    bucket_name     = "infra_logs"
    s3_key_prefix   = "log/redshift-"
  }
}
```

For Amazon Global Accelerator:

```
resource "aws_globalaccelerator_accelerator" "accelerat
  attributes {
    flow_logs_enabled   = true
    flow_logs_s3_bucket = "example-bucket"
    flow_logs_s3_prefix = "flow-logs/"
  }
}
```

For Amazon OpenSearch service, or Amazon Elasticsearch service:

```
resource "aws_elasticsearch_domain" "domain" {
  log_publishing_options {
    cloudwatch_log_group_arn = "arn:aws:logs:us-east-1:
    log_type = "AUDIT_LOGS"
    enabled = true
  }
}
```

For Amazon CloudFront distributions:

```
resource "aws_cloudfront_distribution" "cloudfront_dist
  default_root_object = "index.html"
  logging_config {
    bucket          = "mycompliantbucketname"
    prefix          = "log/cloudfront-"
  }
}
```

For both Amazon Classic Load Balancing and Application Load Balancing:

```
resource "aws_lb" "load_balancer" {
  access_logs {
    enabled = true
    bucket = "mycompliantbucket"
    bucket_prefix = "log/lb-"
  }
}
```

For GCP Cloud Storage service:

```
resource "google_storage_bucket" "example" {
  name     = "example"
  location = "US"
  logging {
    log_bucket = google_storage_bucket.bucket-log.name
  }
}
```

For GCP Region Backend Service:

```
resource "google_compute_region_backend_service" "examp
  name                            = "example"
  region                          = "us-central1"
  health_checks                   = [google_compute_reg
  connection_draining_timeout_sec = 10
  session_affinity                = "CLIENT_IP"
  load_balancing_scheme           = "EXTERNAL"
  protocol                        = "HTTPS"

  log_config {
    enable = true
  }
}
```

For GCP VPC Subnetwork:

```
resource "google_compute_subnetwork" "example" {
  name          = "example"
  ip_cidr_range = "10.2.0.0/16"
  region        = "us-central1"
  network       = google_compute_network.custom-test.id
  log_config {
    aggregation_interval = "INTERVAL_10_MIN"
    flow_sampling        = 0.5
    metadata             = "INCLUDE_ALL_METADATA"
  }
}
```

For GCP SQL Database Instance:

```
resource "google_sql_database_instance" "example" {
  name             = "example"
  database_version = "POSTGRES_11"
  region           = "us-central1"

  settings {
    tier = "db-f1-micro"
    ip_configuration {
      require_ssl  = true
      ipv4_enabled = true
    }
    database_flags {
      name  = "log_connections"
      value = "on"
    }
    database_flags {
      name  = "log_disconnections"
      value = "on"
    }
    database_flags {
      name  = "log_checkpoints"
      value = "on"
    }
    database_flags {
      name  = "log_lock_waits"
      value = "on"
    }
  }
}
```

For GCP Kubernetes Engine (GKE) cluster:

```
resource "google_container_cluster" "example" {
  name               = "example"
  location           = "us-central1-a"
  initial_node_count = 3
  logging_service    = "logging.googleapis.com/kubernet
}
```

See

- OWASP Top 10 2021 Category A9 - Security Logging and Monitoring Failures
- AWS Documentation - Logging requests using server access logging
- MITRE, CWE-778 - Insufficient Logging
- OWASP Top 10 2017 Category A10 - Insufficient Logging & Monitoring

Available In:

sonarcloud | sonarqube