Code like Hacker: Secure .tf Practices

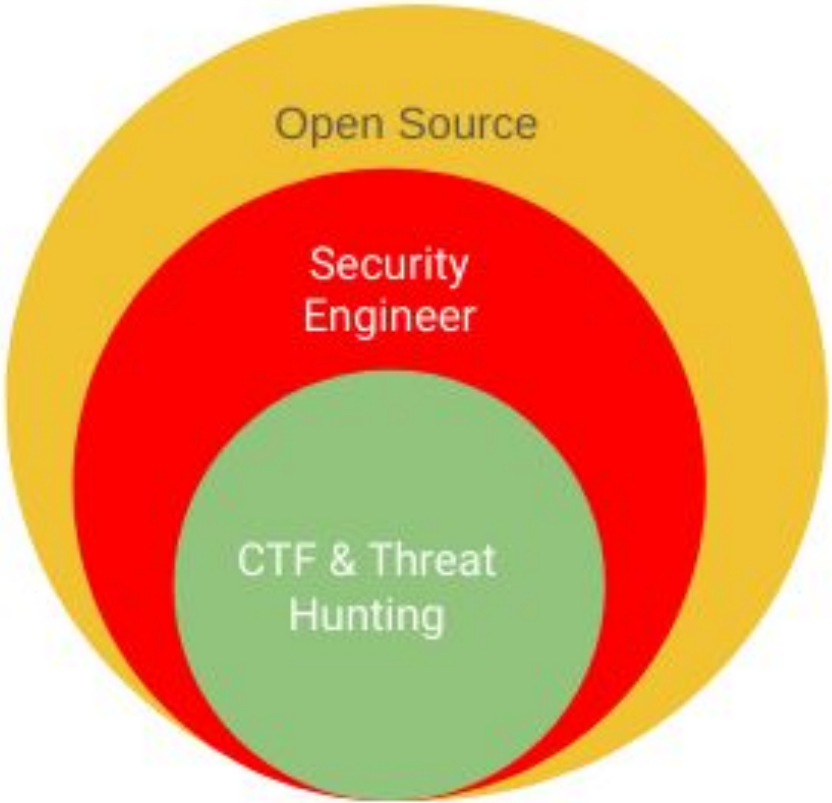
Art of writing secure codes

$whoami

Shubhendu Shubham

“sudo rm -rf / problems”

aka “Troubleshooter”



CTF BADGES

CERTIFICATIONS

SC 100

AZ 305

AZ 104

AZ 500

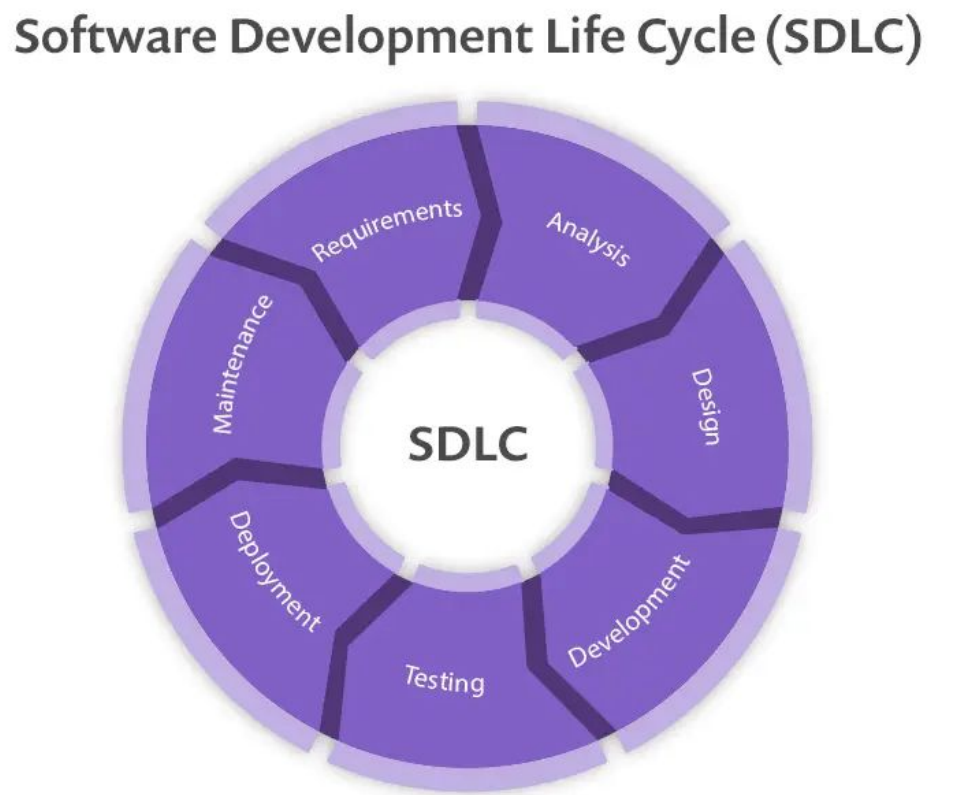
AZ 700

Community

Disclaimer

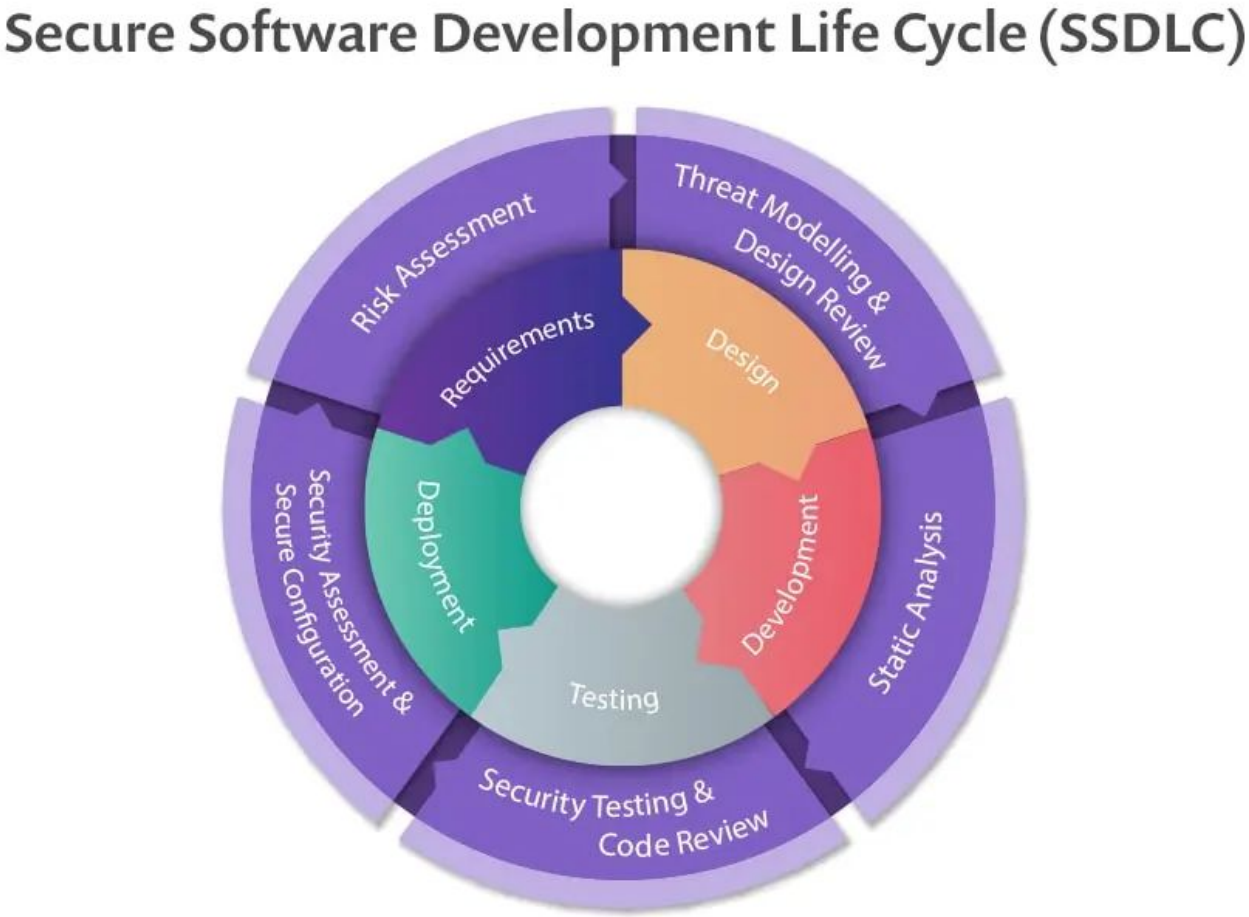
“A good programmer is someone who always looks both ways before crossing a one-way street.” — Doug Linder

7 Phases of SDLC

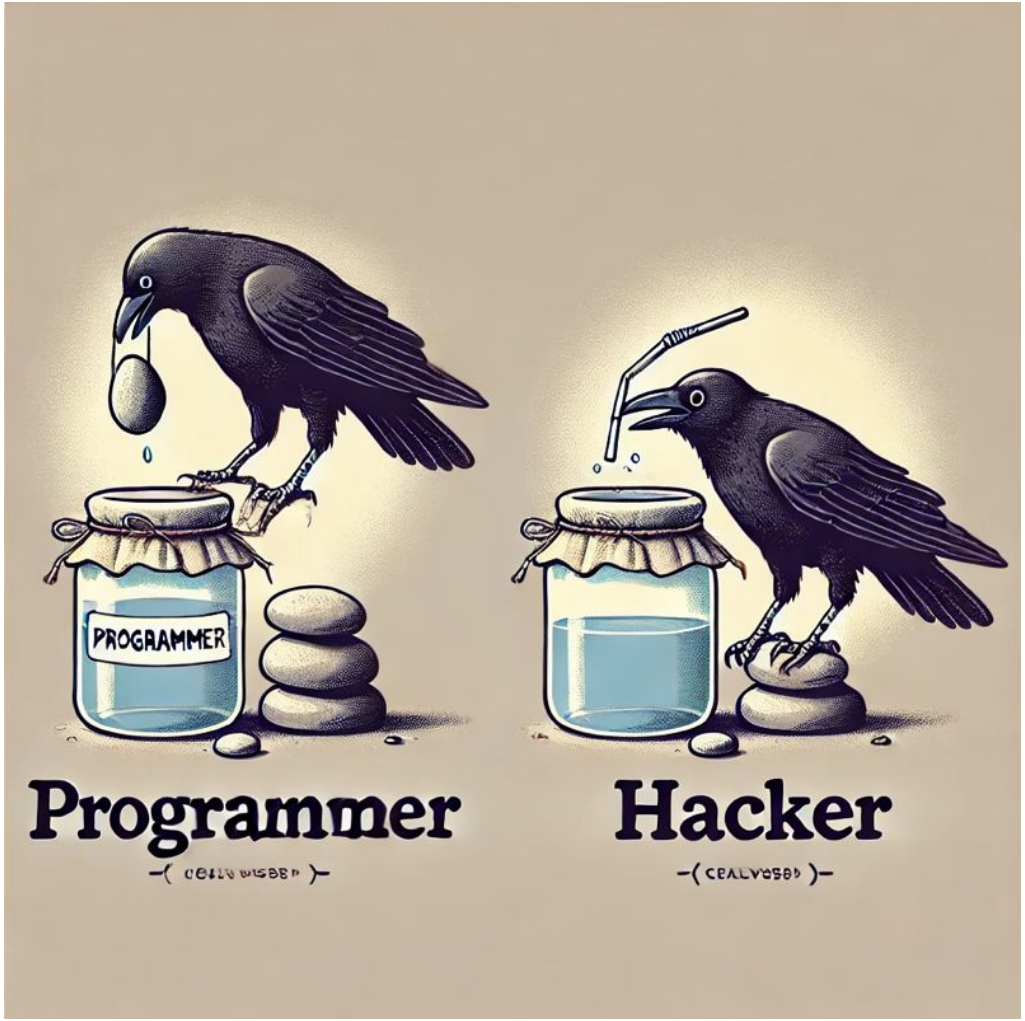


snyk

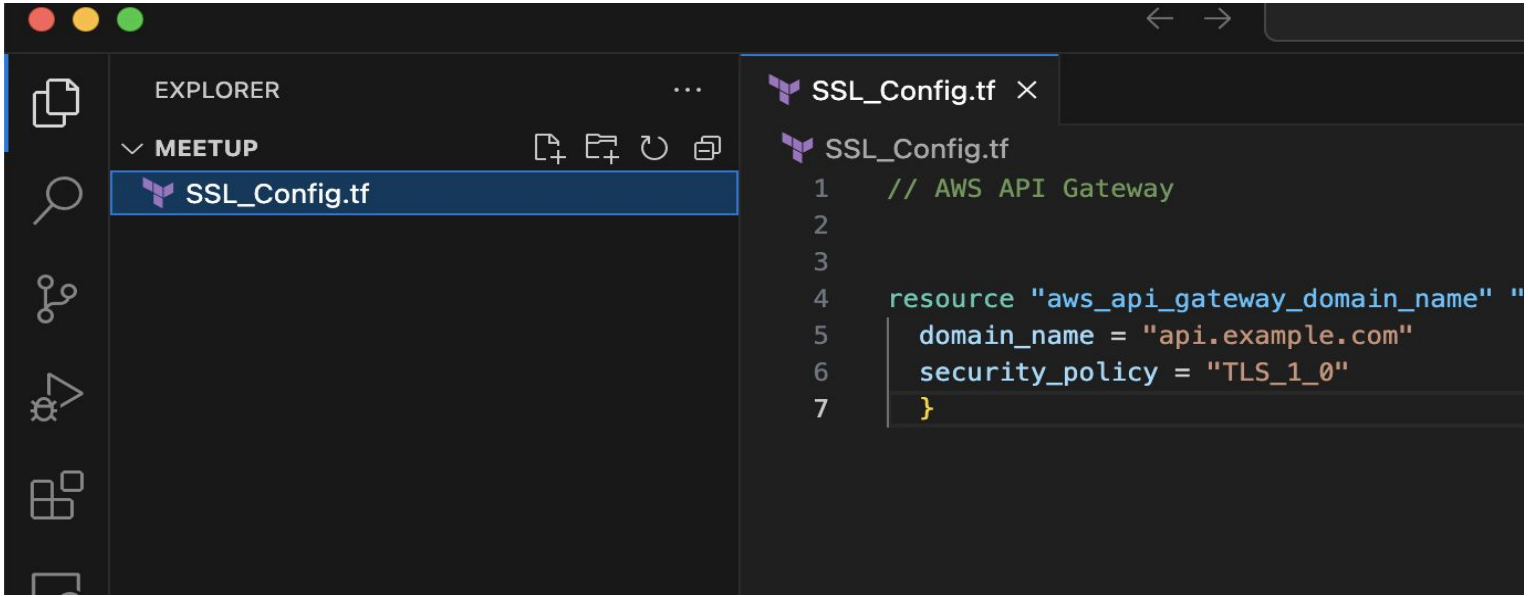
5 Phases of SSDLC



snyk


PROGRAMMER
Programmer
Hacker

SSL/TLS Protocol



SSL/TLS protocols

```
8
9   // Compliant
0
1     resource "aws_api_gateway_domain_name" "example" {
2     domain_name = "api.example.com"
3     security_policy = "TLS_1_2"
4   }
```

Compliant

Compliant SSL/TLS protocols

```
resource "aws_elasticsearch_domain" "example" {
  domain_name = "example"
  domain_endpoint_options {
    enforce_https = true
    tls_security_policy = "Policy-Min-TLS-1-0-2019-07"
  }
}
```

Compliant SSL/TLS protocols

```
resource "aws_elasticsearch_domain" "example" {
  domain_name = "example"
  domain_endpoint_options {
    enforce_https = true
    tls_security_policy = "Policy-Min-TLS-1-2-2019-07"
  }
}
```

Compliant

Policies Authorizing Public Access to Resources

```
  bucket = aws_s3_bucket.mybucket.id
  policy = jsonencode({
    Id = "mynoncompliantpolicy"
    Version = "2012-10-17"
    Statement = [{
          Effect = "Allow"
          Principal = {
              AWS = "*"
          }
          Action = [
              "s3:PutObject"
          ]
          Resource: "${aws_s3_bucket.mybucket.arn}/*"
      }
    ]
  })
}
```

Policies Authorizing Public Access to Resources

```
resource "aws_s3_bucket_policy" "mycompliantpolicy" {
  bucket = aws_s3_bucket.mybucket.id
  policy = jsonencode({
    Id = "mycompliantpolicy"
    Version = "2012-10-17"
    Statement = [{
          Effect = "Allow"
          Principal = {
            AWS = [
              "arn:aws:iam::${data.aws_caller_identity.current.account_id}:root"
            ]
          }
          Action = [
            "s3:PutObject"
          ]
          Resource = "${aws_s3_bucket.mybucket.arn}/*"
        }
    ]
  })
}
```

Compliant

Administration services access should be restricted to specific IP

```
resource "aws_security_group" "noncompliant" {
  name        = "allow_ssh_noncompliant"
  description = "allow_ssh_noncompliant"
  vpc_id      = aws_vpc.main.id

  ingress {
    description      = "SSH rule"
    from_port        = 22
    to_port          = 22
    protocol         = "tcp"
    cidr_blocks      = ["0.0.0.0/0"]  # Noncompliant
  }
}
```

Administration services access should be restricted to specific IP

```
resource "aws_security_group" "compliant" {
  name        = "allow_ssh_compliant"
  description = "allow_ssh_compliant"
  vpc_id      = aws_vpc.main.id

  ingress {
    description      = "SSH rule"
    from_port        = 22
    to_port          = 22
    protocol         = "tcp"
    cidr_blocks      = ["1.2.3.0/24"]
  }
}
```

Compliant

Disabling Logging : Security-Sensation

```
resource "azurerm_app_service" "example" {
  name                = "example-app-service"
  location            = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  app_service_plan_id = azurerm_app_service_plan.example.id

  logs {
    application_logs {
      file_system_level = "Off"
      azure_blob_storage {
        level = "Off"
        sas_url = azurerm_storage_account.example.primary_blob_endpoint
        retention_in_days = 7
      }
    }
  }
}
```

Disabling Logging : Security-Sensation

```
resource "azurerm_app_service" "example" {
  name                = "example-app-service"
  location            = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  app_service_plan_id = azurerm_app_service_plan.example.id

  logs {
    http_logs {
      file_system {
        retention_in_days = 90  # Retain logs for 90 days
        retention_in_mb   = 100 # Limit log storage to 100 MB
      }
    }

    application_logs {
      file_system_level = "Error" # Log only errors to the file system

      azure_blob_storage {
        sas_url           = azurerm_storage_account.example.primary_blob_endpoint # SAS URL required for Blob Storage
        retention_in_days = 90     # Retain logs for 90 days
        level             = "Error" # Log only errors to Blob Storage
      }
    }
  }
}This Terraform code snippet demonstrates best practices and potential vulnerabilities related to security configurations a
```

Compliant

Disabling Logging : Security-Sensation

File System Logging:

```hcl
file_system_level = "Off" # Sensitive
```

- `file_system_level`:
  - Specifies whether application logs should be stored in the **file system**.
  - Possible values: `"Off"`, `"Verbose"`, `"Error"`, `"Information"`, `"Warning"`.
  - `"Off"` means no logs are written to the file system.
- **Sensitive**:
  - Application logs might contain sensitive information, so logging to the file system is disabled here to enhance security.

Disabling S3 bucket MFA delete : Security-Sentiton

```
//A versioned S3 bucket does not have MFA delete enabled for AWS provider version 3 or below

resource "aws_s3_bucket" "example" { # Sensitive
  bucket = "example"

  versioning {
    enabled = true
  }
}

//A versioned S3 bucket does not have MFA delete enabled for AWS provider version 4 or above:

resource "aws_s3_bucket" "example" {
  bucket = "example"
}

resource "aws_s3_bucket_versioning" "example" { # Sensitive
  bucket = aws_s3_bucket.example.id
  versioning_configuration {
    status = "Enabled"
  }
}
```

Disabling S3 bucket MFA delete : Security-Sentiton

```
resource "aws_s3_bucket" "example" {
  bucket = "example"

  versioning {
    enabled = true
    mfa_delete = true
  }
}


//A versioned S3 bucket does not have MFA delete enabled for AWS provider version 4 or above:


resource "aws_s3_bucket" "example" {
  bucket = "example"
}

resource "aws_s3_bucket_versioning" "example" {
  bucket = aws_s3_bucket.example.id
  versioning_configuration {
    status = "Enabled"
    mfa_delete = "Enabled"
  }
  mfa = "${var.MFA}"
}
```

Compliant

Tips

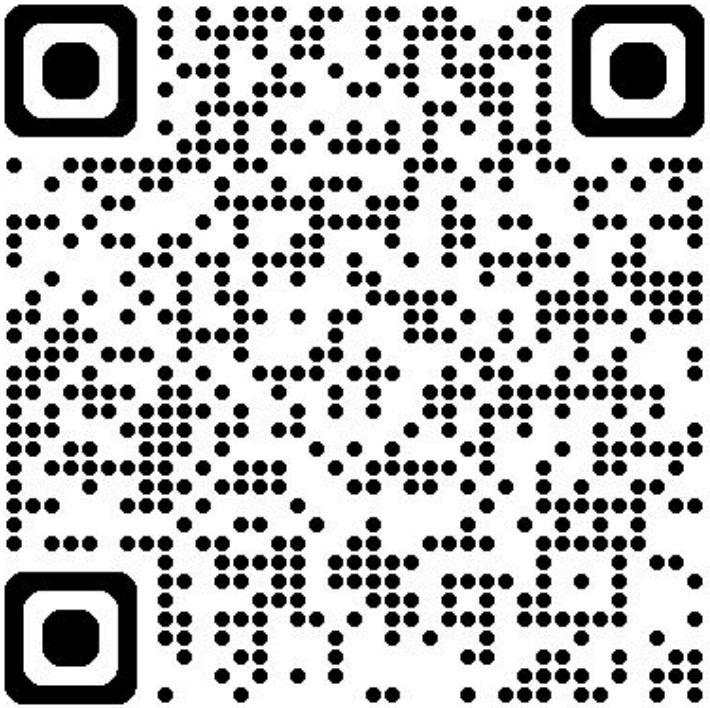
- MFA delete can only be enabled with the AWS CLI or API and with the root account
- To delete an object version, the API should be used with the *x-amz-mfa* header
- The API request, with the *x-amz-mfa* header, can only be used in HTTPS

References

1. https://snyk.io/learn/secure-sdlc/
2. https://rules.sonarsource.com/terraform/
3. https://www.tenable.com/terrascan
4. https://github.com/sivolko/teeraform_SAST

What's Next ?

This is not Phishing QR ,It's my LinkedIn **Don't Trust** Always verify



Thank you!