Infrastructure-as-Code (IaC) SAST: What's that?

and why IaC SAST?

Chaitra Bhat
Security Engineer @Yahoo!

yahoo!

About Me

Chaitra Bhat

Security Engineer, a Paranoid @Yahoo!

Long time developer, now a full time security engineer

- LinkedIn: https://www.linkedin.com/in/bhatchaitra/
- Twitter: @cyberbhatc

yahoo!

Let’s get some definitions out of the way

What is Infrastructure-as-Code (IaC)?

Machine-readable definitions in the form of code or script to manage and provision your infrastructure rather than doing it manually.

What is Static application security testing (SAST)?

SAST or static analysis is a testing methodology that analyzes source code to find security vulnerabilities in the code.

Objectives

What?

What is IaC SAST? Some examples

Why?

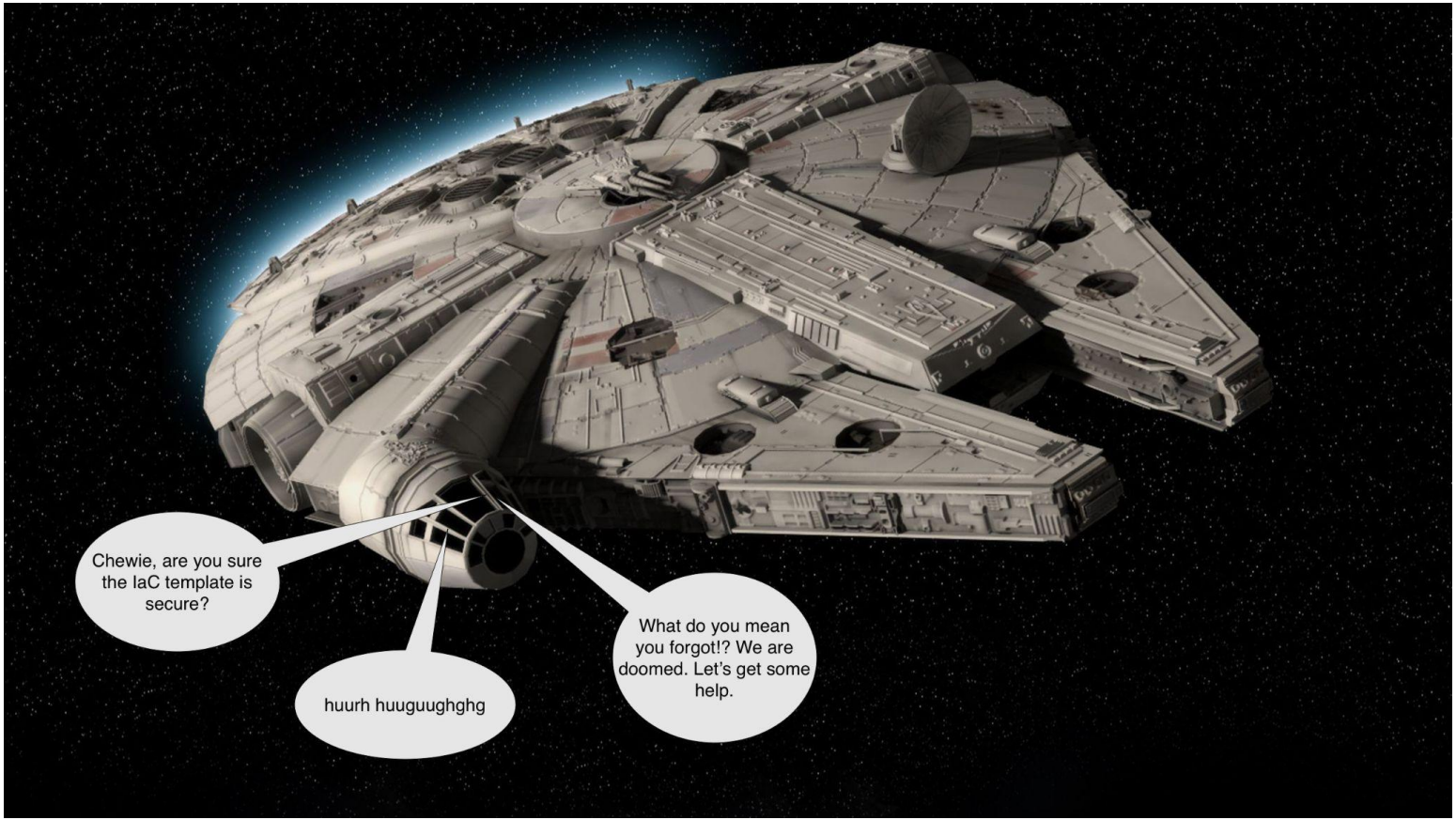
Why does one need IaC SAST tools?

Where?

Where does IaC SAST fit in DevSecOps?

IaC all the way!

So, we are FTL... what can possibly go wrong?



What's the solution?

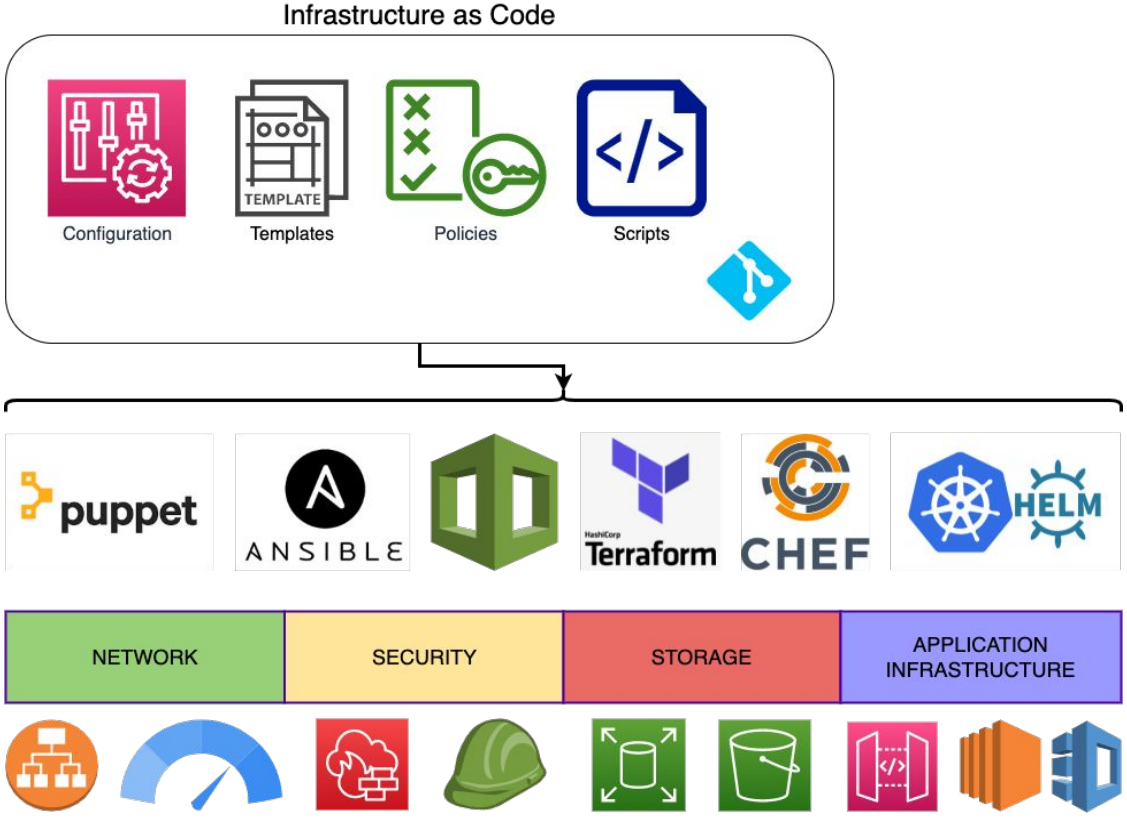
Outline

1. IaC
2. Vulnerabilities in infrastructure
3. IaC SAST
4. IaC SAST tools
5. IaC SAST in DevSecOps
6. Conclusion

Outline

1. IaC
2. Vulnerabilities in infrastructure
3. IaC SAST
4. IaC SAST tools
5. IaC SAST in DevSecOps
6. Conclusion

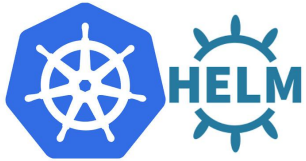
IaC (what is)


Infrastructure as Code
Configuration
TEMPLATE
Templates
Policies
Scripts
puppet
ANSIBLE
HashiCorp
Terraform
CHEF
HELM
NETWORK
SECURITY
STORAGE
APPLICATION INFRASTRUCTURE

IaC (2)

Examples of IaC tools/services

Infrastructure Provisioning and Management Tools



Configuration Management Tools

IaC (3)

What are the benefits of IaC?

Automation

Automated workflows integrated with the CI/CD pipeline

Repeatability

Code templates facilitate repeatability

Scalability

Easy resource management and provisioning makes it scalable

Security

Secure templates resulting in secure infrastructure

These benefits can so easily turn into pitfalls!

Outline

1. IaC
2. Vulnerabilities in infrastructure
3. IaC SAST
4. IaC SAST tools
5. IaC SAST in DevSecOps
6. Conclusion

Vulnerabilities in Infrastructure (1)

What are the potential vulnerabilities in the infrastructure?

Misconfigurations in IaC can result in vulnerabilities in the infrastructure

Vulnerability Type	Potential misconfigurations
Network exposure	Open security groups, publicly accessible cloud storage services, public ssh access, databases that are accessible from the internet
Unauthorised privilege escalations	Incorrectly stored secrets, containers running as root
Improper access control	Excessive permissions to your resources
Insufficient & Insecure logging	Logging not enabled, logs not encrypted

Vulnerabilities in Infrastructure (2)

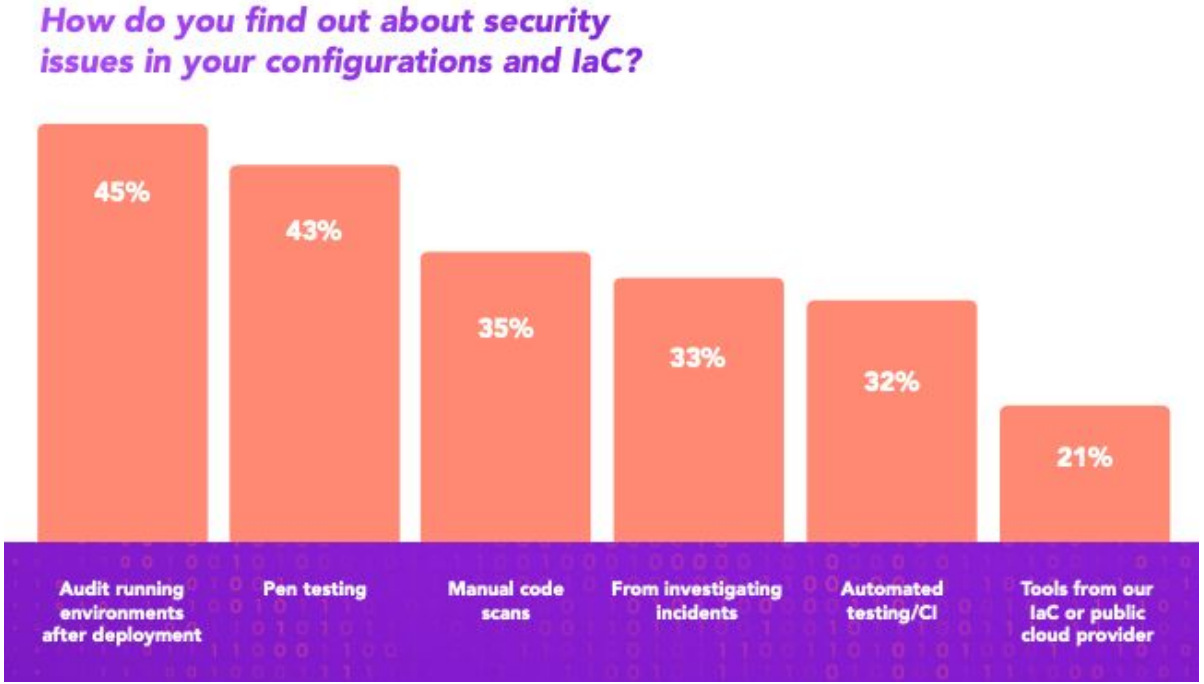
What causes these misconfigurations in IaC?

- Oversight by the engineer
- New to IaC
- Complex IaC configurations
- Fast moving devops cycles
- Gaps in testing
- Dependency on pen-testing
- Dependency on dynamic testing

It's that easy to cause vulnerabilities in your infrastructure!

Vulnerabilities in Infrastructure (4)

Industry reports on IaC misconfigurations (1)



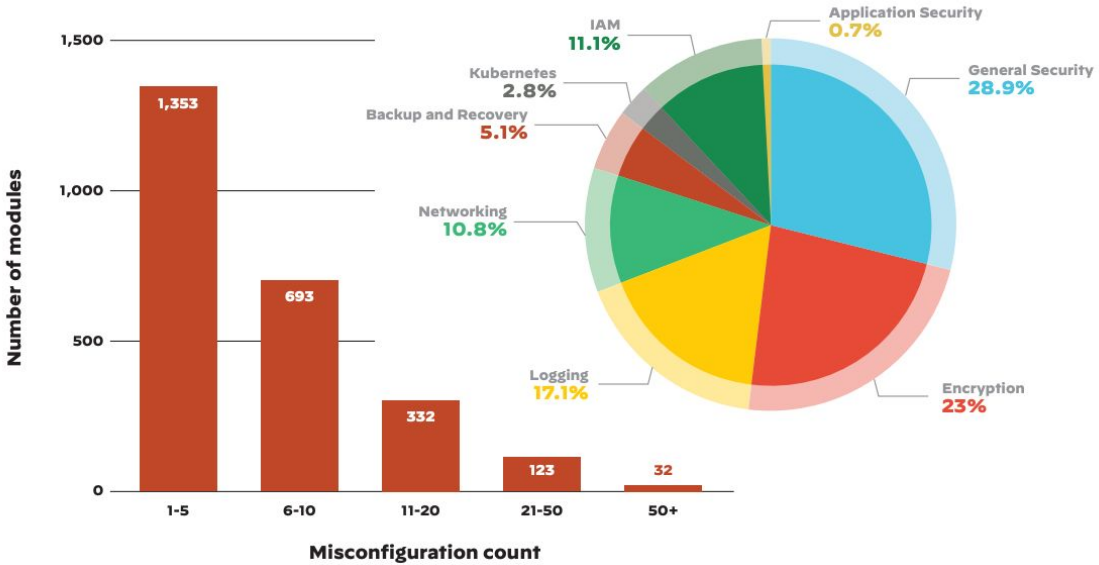
Security issues found too late in the workflow: As per [2021 IaC Research Report] from Synk, **45%** of IaC misconfigurations were found after deployment.

Post-deployment checks take

- > 1 week to discover a security issue (if discoverable)
- > 1 day to fix those issues

Vulnerabilities in Infrastructure (3)

Industry reports on IaC misconfigurations (2)



Public Terraform modules by number of misconfigurations (left); types of misconfigurations and their percentages (right)

IaC key to Supply Chain Protection: 2021 Cloud Threat Report from Unit 42 of Palo Alto Networks shows how including various* Terraform modules increases the chances of misconfigurations in IaC.

Note: IaC misconfigurations are created by the **cloud user**, not by CSPs or IaC providers. Hence, in the context of the **shared responsibility model**, it is important to secure the IaC templates.

* 4,055 Terraform templates and 38,480 Terraform files in popular open-source Terraform repositories were analysed

Vulnerabilities in Infrastructure (5)

Some examples of misconfigurations in IaC from TerraGoat project

```
resource "aws_security_group" "web-node" {
  # security group is open to the world in SSH port
  name        = "${local.resource_prefix.value}-sg"
  description = "${local.resource_prefix.value} Security Group"
  vpc_id      = aws_vpc.web_vpc.id

  ingress {
    from_port = 80
    to_port   = 80
    protocol  = "tcp"
    cidr_blocks = [
    "0.0.0.0/0"]
  }
  ingress {
    from_port = 22
    to_port   = 22
    protocol  = "tcp"
    cidr_blocks = [
    "0.0.0.0/0"]
  }
```

Ex1: SSH port is open for all in the Security Group

```
resource "aws_s3_bucket" "flowbucket" {
  bucket        = "${local.resource_prefix.value}-flowlogs"
  force_destroy = true

  tags = merge({
    Name        = "${local.resource_prefix.value}-flowlogs"
    Environment = local.resource_prefix.value
    }, {
    git_commit           = "d68d2897add9bc2203a5ed0632a5cdd8ff8cefb0"
    git_file             = "terraform/aws/ec2.tf"
    git_last_modified_at = "2020-06-16 14:46:24"
    git_last_modified_by = "nimrodkor@gmail.com"
    git_modifiers        = "nimrodkor"
    git_org              = "bridgecrewio"
    git_repo             = "terragoat"
    yor_trace            = "f058838a-b1e0-4383-b965-7e06e987ffb1"
  })
}
```

Ex2: S3 bucket is not encrypted at rest

Vulnerabilities in Infrastructure (6)

Some more examples of misconfigurations in IaC...

```
resource "aws_s3_bucket" "flowbucket" {
  bucket        = "${local.resource_prefix.value}-flowlogs"
  force_destroy = true

  tags = merge({
    Name        = "${local.resource_prefix.value}-flowlogs"
    Environment = local.resource_prefix.value
    }, {
    git_commit           = "d68d2897add9bc2203a5ed0632a5cdd8ff8cefb0"
    git_file             = "terraform/aws/ec2.tf"
    git_last_modified_at = "2020-06-16 14:46:24"
    git_last_modified_by = "nimrodkor@gmail.com"
    git_modifiers        = "nimrodkor"
    git_org              = "bridgecrewio"
    git_repo             = "terragoat"
    yor_trace            = "f058838a-b1e0-4383-b965-7e06e987ffb1"
  })
}
```

Ex3: Logging for access to S3 bucket not enabled

```
data aws_iam_policy_document "policy" {
  statement {
    actions = ["es:*"]
    principals {
      type        = "AWS"
      identifiers = ["*"]
    }
    resources = ["*"]
  }
}
```

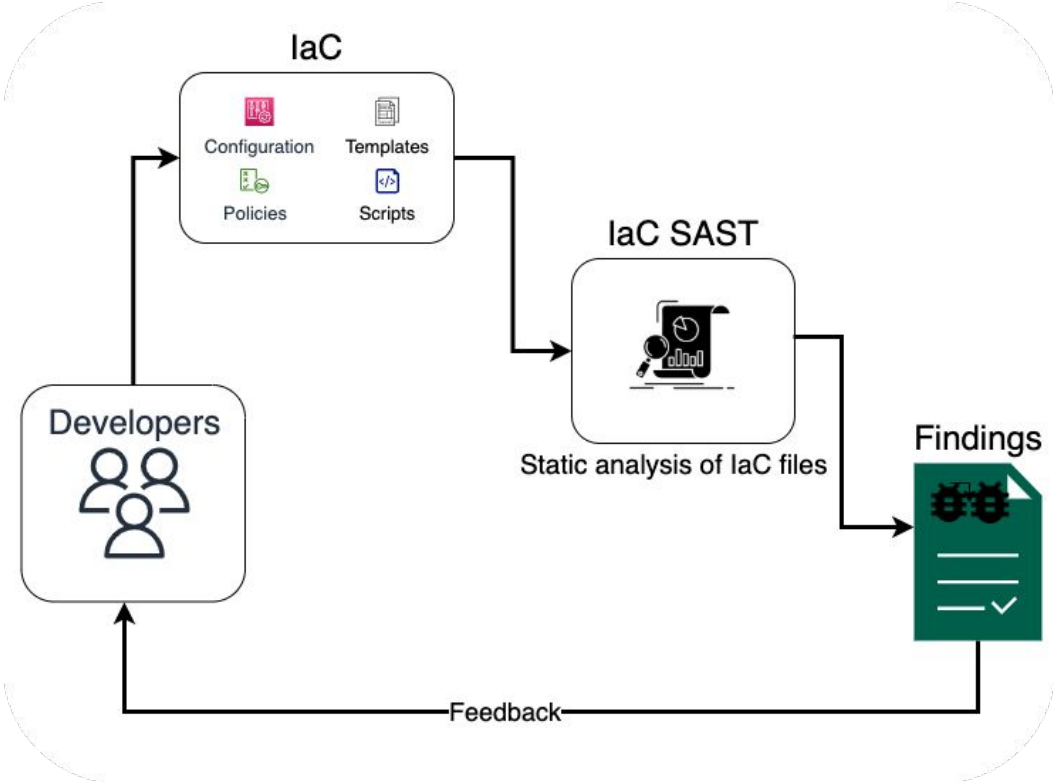
Ex4: Fine grained access control for resources not defined

Have I scared you enough?

Topics

1. IaC
2. Vulnerabilities in infrastructure
3. IaC SAST
4. IaC SAST tools
5. IaC SAST in DevSecOps
6. Conclusion

IaC SAST (what is)



Static analysis of the IaC files to find security misconfigurations in the code.

IaC SAST (1)

Identifying vulnerabilities in IaC using Checkov IaC SAST tool as example

```
Check: CKV_AWS_24: "Ensure no security groups allow ingress from 0.0.0.0:0 to port 22"
        FAILED for resource: aws_security_group.web-node
        File: /ec2.tf:77-115
        Guide: https://docs.bridgecrew.io/docs/networking_1-port-security
```

Ex1: SSH port is open for all in the Security Group

```
Check: CKV_AWS_19: "Ensure all data stored in the S3 bucket is securely encrypted at rest"
        FAILED for resource: aws_s3_bucket.flowbucket
        File: /ec2.tf:271-288
        Guide: https://docs.bridgecrew.io/docs/s3_14-data-encrypted-at-rest
```

Ex2: S3 bucket is not encrypted at rest

Not promoting any tool, just using them as examples.

IaC SAST (2)

Identifying vulnerabilities in IaC using Checkov as example

```
Check: CKV_AWS_18: "Ensure the S3 bucket has access logging enabled"
        FAILED for resource: aws_s3_bucket.flowbucket
        File: /ec2.tf:271-288
        Guide: https://docs.bridgecrew.io/docs/s3_13-enable-logging
```

Ex3: Logging for access to S3 bucket not enabled

```
Check: CKV_AWS_111: "Ensure IAM policies does not allow write access without constraints"
       FAILED for resource: aws_iam_policy_document.policy
       File: /es.tf:29-38
       Guide: https://docs.bridgecrew.io/docs/ensure-iam-policies-do-not-allow-write-access-without-constraint
```

Ex4: Fine grained access control for resources not defined

Topics

1. IaC
2. Vulnerabilities in infrastructure
3. IaC SAST
4. IaC SAST tools
5. IaC SAST in DevSecOps
6. Conclusion

IaC SAST Tools (1)

Some examples of IaC SAST tools in the market today

checkov by bridgecrew

kics. by Checkmarx

snyk

aqua tfsec

cloudrail

accurics by tenable

Not promoting any tool, just using them as examples.

IaC SAST Tools (2)

Some criteria when **selecting an IaC SAST tool**

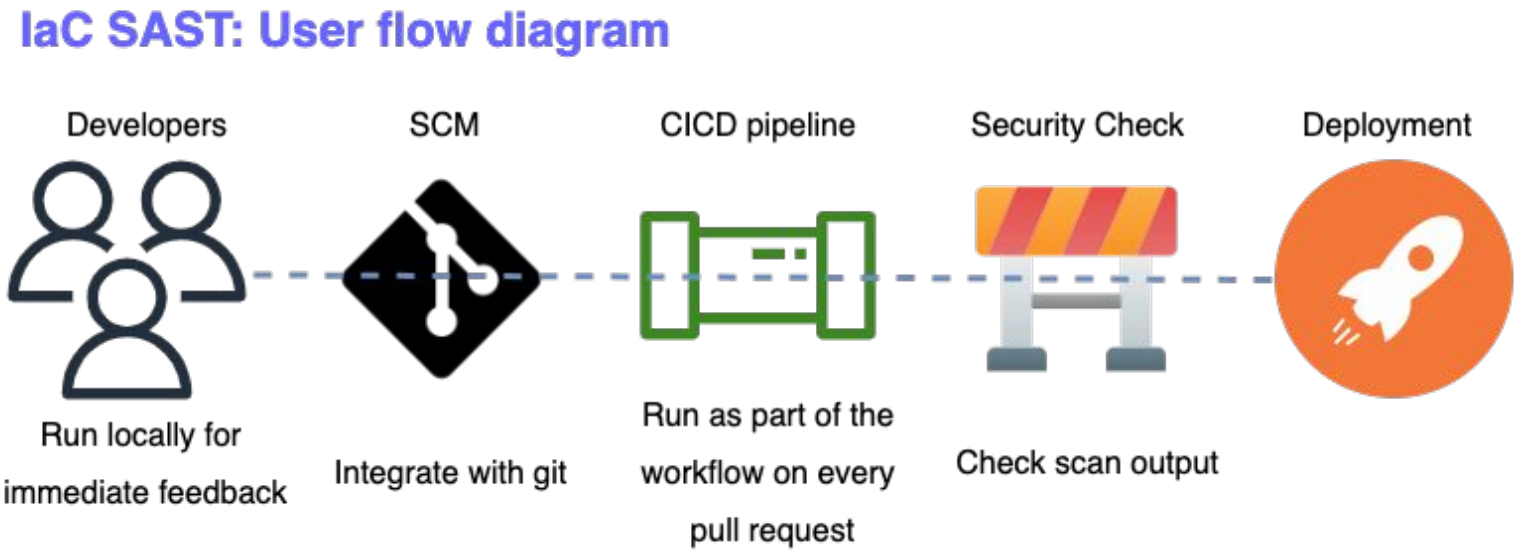
- **File types** supported: Terraform, Cloudformation, Docker, Kubernetes, etc
- **Benchmarks** supported: CIS, NIST
- Ability to **integrate with SCM** systems like Git for automated workflows
- **Output format** supported: JSON, HTML
- **High SNR** (signal-to-noise ratio) or low false positives
- Ability to **understand the context** and resolve references in IaC
- Ability to **add/customise** checks/rulesets
- **OSS** or **paid** version
- Ability to check runtime environment to **identify configuration drifts**

Topics

1. IaC
2. Vulnerabilities in infrastructure
3. IaC SAST
4. IaC SAST tools
5. IaC SAST in DevSecOps
6. Conclusion

IaC SAST integration (1)

How and where can IAC SAST tool be integrated?



IaC SAST integration (2)

Some tips for **efficient integration** of IaC SAST tool

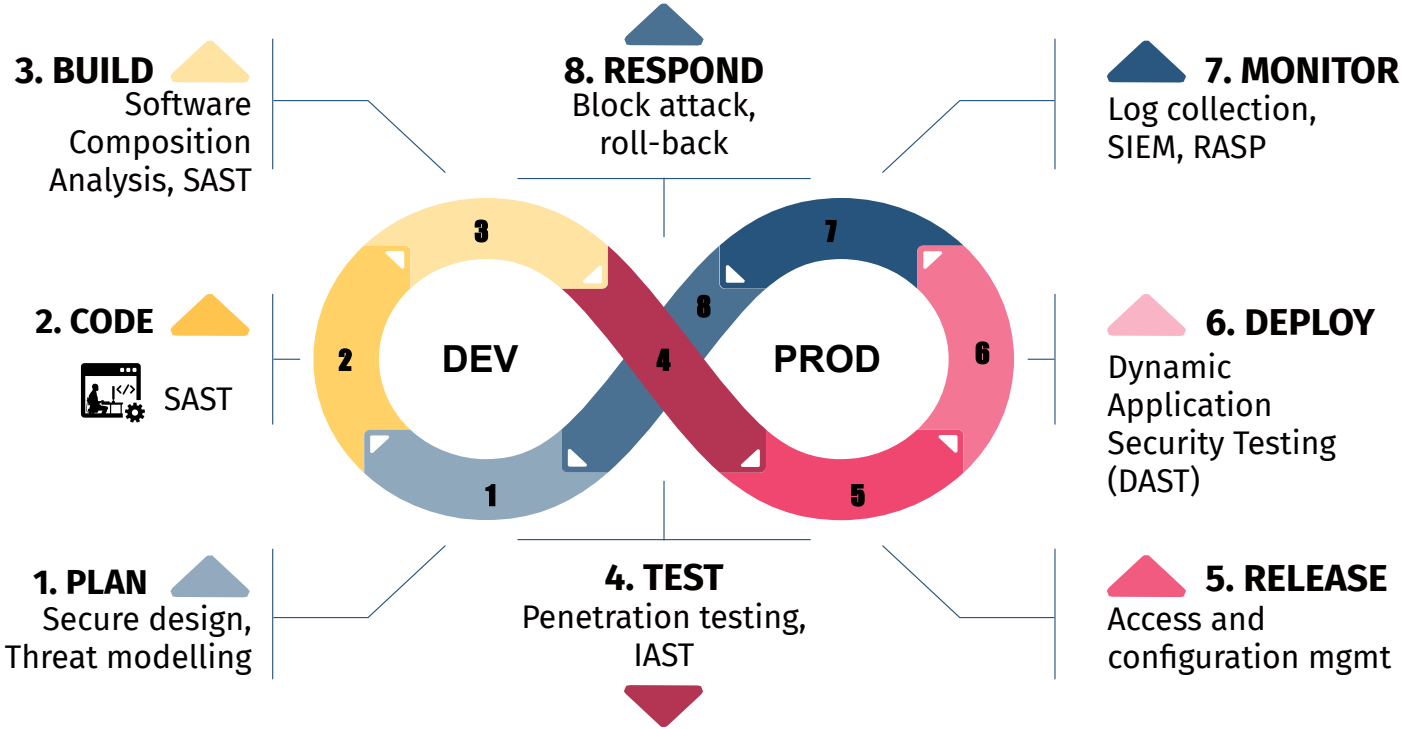
- Integration with IDE, SCM like GitHub, GitLab, for immediate feedback
- Integration into CI/CD, for workflow automation
- High signal-to-noise ratio on the findings

  I was able to reduce the number of findings from **143 to 16** on one of our internal projects by passing **high value security checks** only to the IaC SAST tool.
- Effective false positive management
- Low-friction security experience for developers

Important to find the balance it is, between the security risk and the value flow!

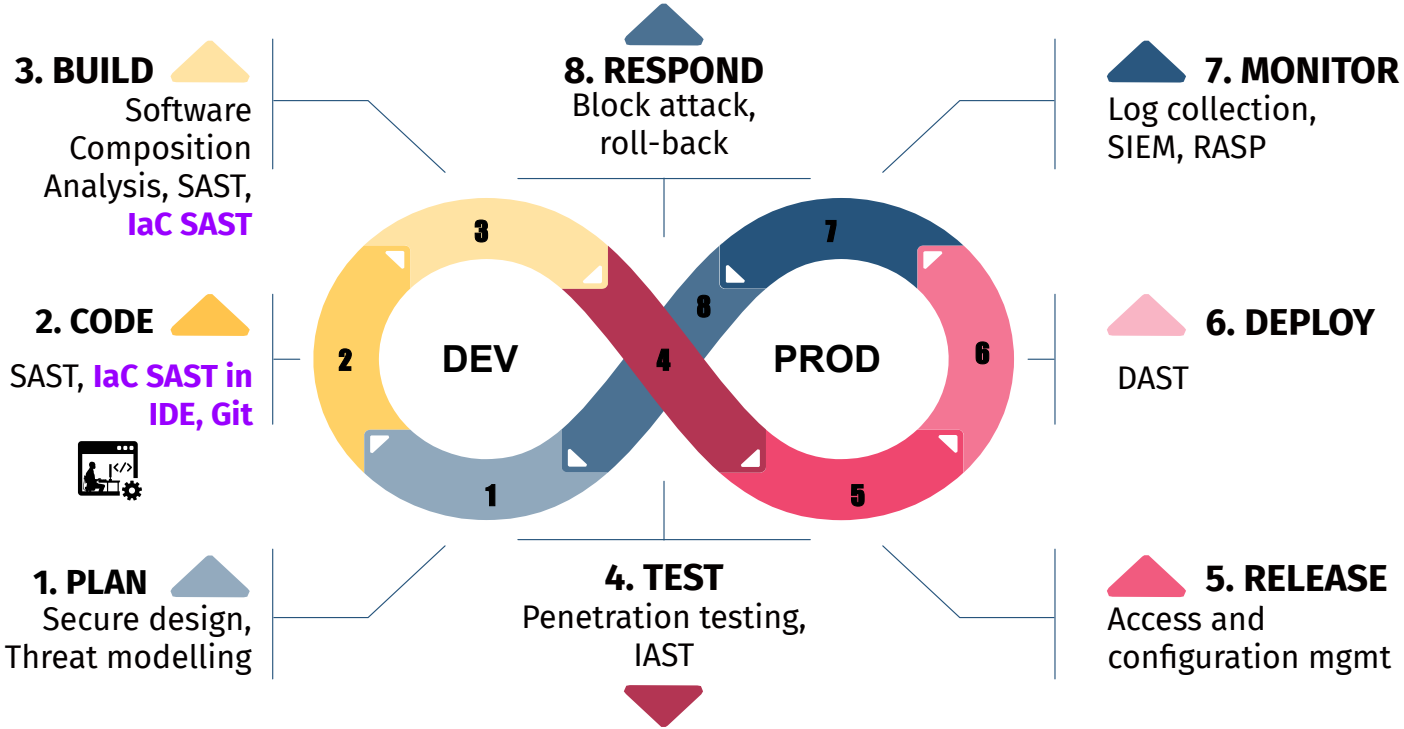
IaC SAST integration (3)

Typical DevSecOps Cycle



IaC SAST integration (4)

Typical DevSecOps Cycle with IaC SAST

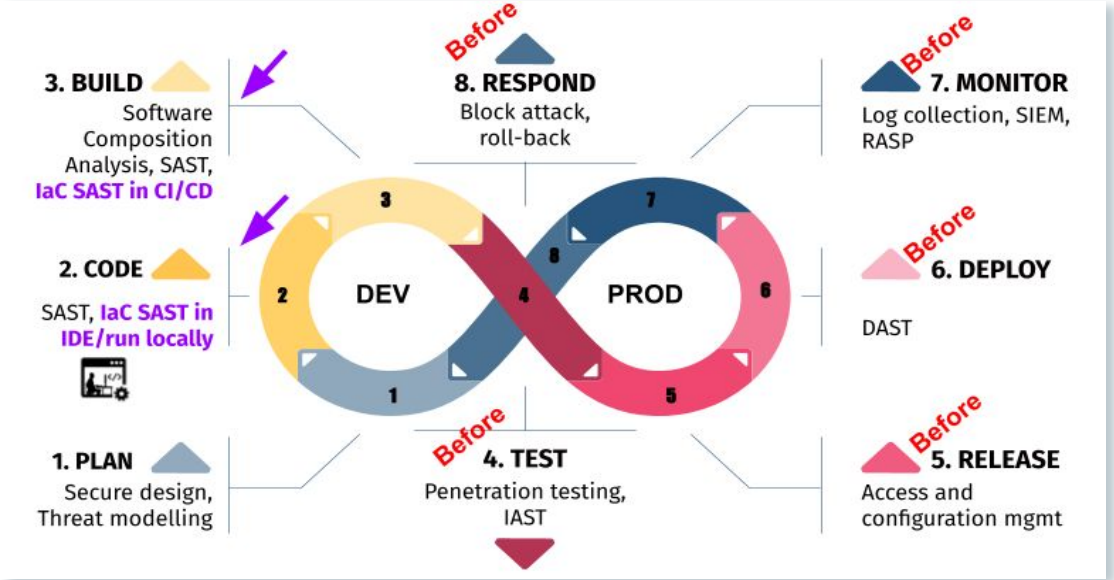


Topics

1. IaC
2. Vulnerabilities in infrastructure
3. IaC SAST
4. IaC SAST tools
5. IaC SAST in DevSecOps
6. Conclusion

Conclusion

IaC SAST Benefits



- Shift-left = less effort, less money & less stress
- Find vulnerabilities before deployment
- Closer to dev cycle
- Helps develop secure mindset
- Secure templates/ paved road
- More information in the code
- Complement DAST findings
- Complement pentesting
- Need not be security experts

Why wait, use the force!

Thank You!