# Introduction to Secure DevOps

DevSecOps

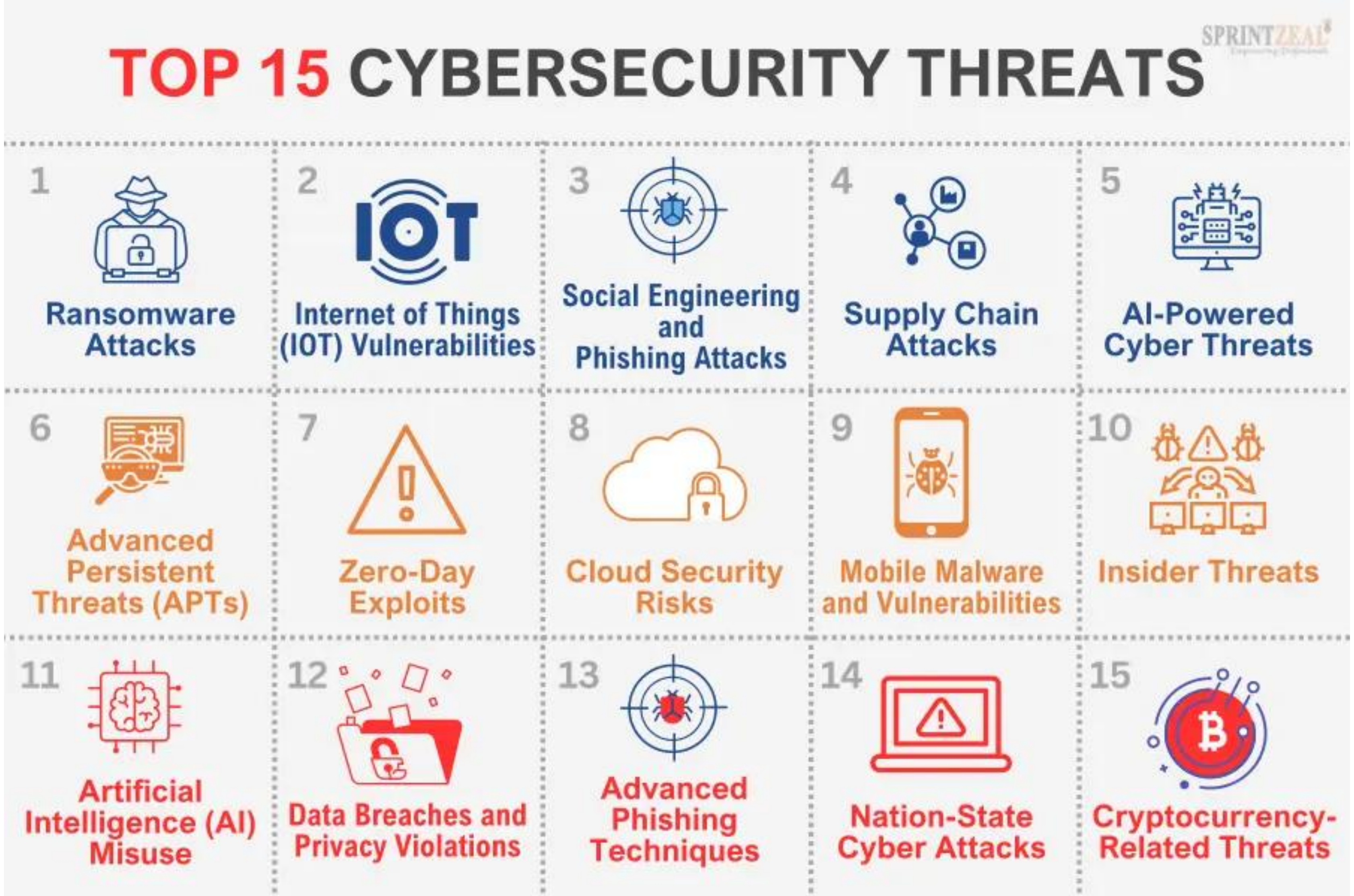
# Agenda

- Security in DevOps
- Principles
- Main Practices

# Security in DevOps

Secure DevOps

# Cybersecurity Threats

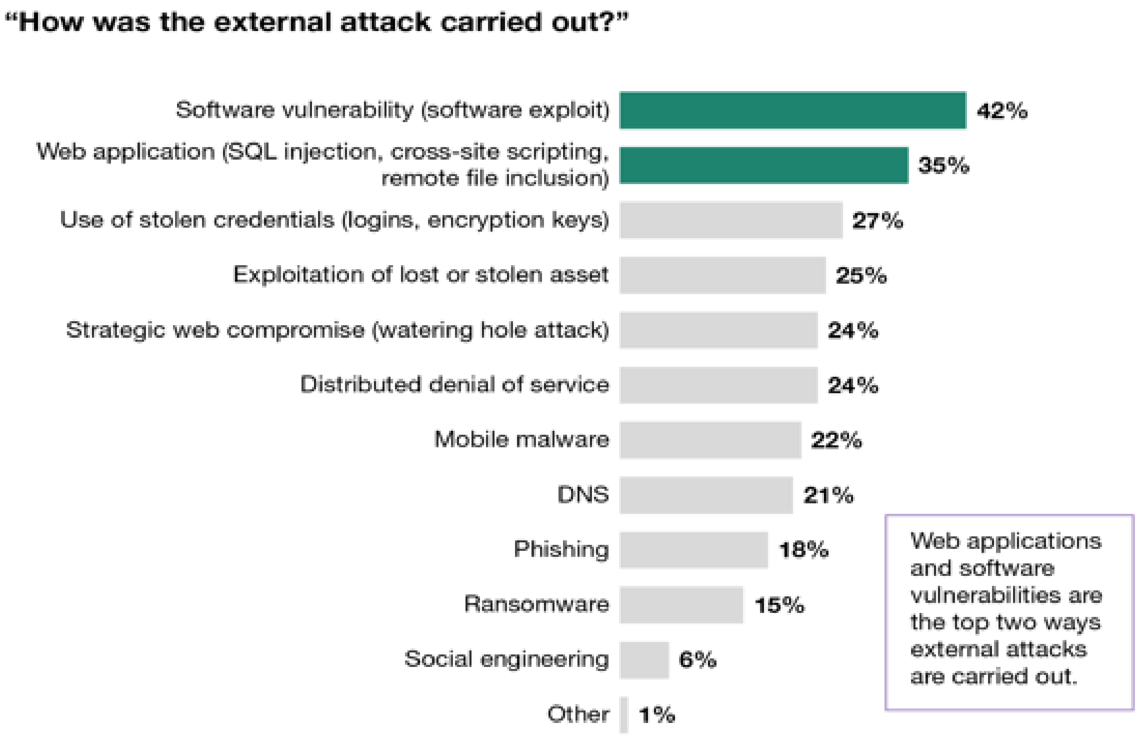


## TOP 15 CYBERSECURITY THREATS

1. Ransomware Attacks
2. Internet of Things (IOT) Vulnerabilities
3. Social Engineering and Phishing Attacks
4. Supply Chain Attacks
5. AI-Powered Cyber Threats
6. Advanced Persistent Threats (APTs)
7. Zero-Day Exploits
8. Cloud Security Risks
9. Mobile Malware and Vulnerabilities
10. Insider Threats
11. Artificial Intelligence (AI) Misuse
12. Data Breaches and Privacy Violations
13. Advanced Phishing Techniques
14. Nation-State Cyber Attacks
15. Cryptocurrency-Related Threats

# Applications as attack vector

- Applications remains the most common attack vector
- Implementing security means be secure in several layers to make it harder to be breached
- Crucial to understand and control all attack vectors
- “You are only as secure as your weakest link”



Web applications and software vulnerabilities are the top two ways external attacks are carried out.

# Assume Breach!

> "FUNDAMENTALLY, IF SOMEBODY WANTS TO GET IN, THEY'RE GETTING IN...ACCEPT THAT.
>
> WHAT WE TELL CLIENTS IS:
> NUMBER ONE, YOU'RE IN THE FIGHT, WHETHER YOU THOUGHT YOU WERE OR NOT. NUMBER TWO,
>
> YOU ALMOST CERTAINLY ARE PENETRATED."

Michael Hayden
Former Director of NSA & CIA

# Assume Breach!

"There are only two types of companies:
those that have been hacked,
and those that will be."

Robert Mueller
FBI Director, 2012

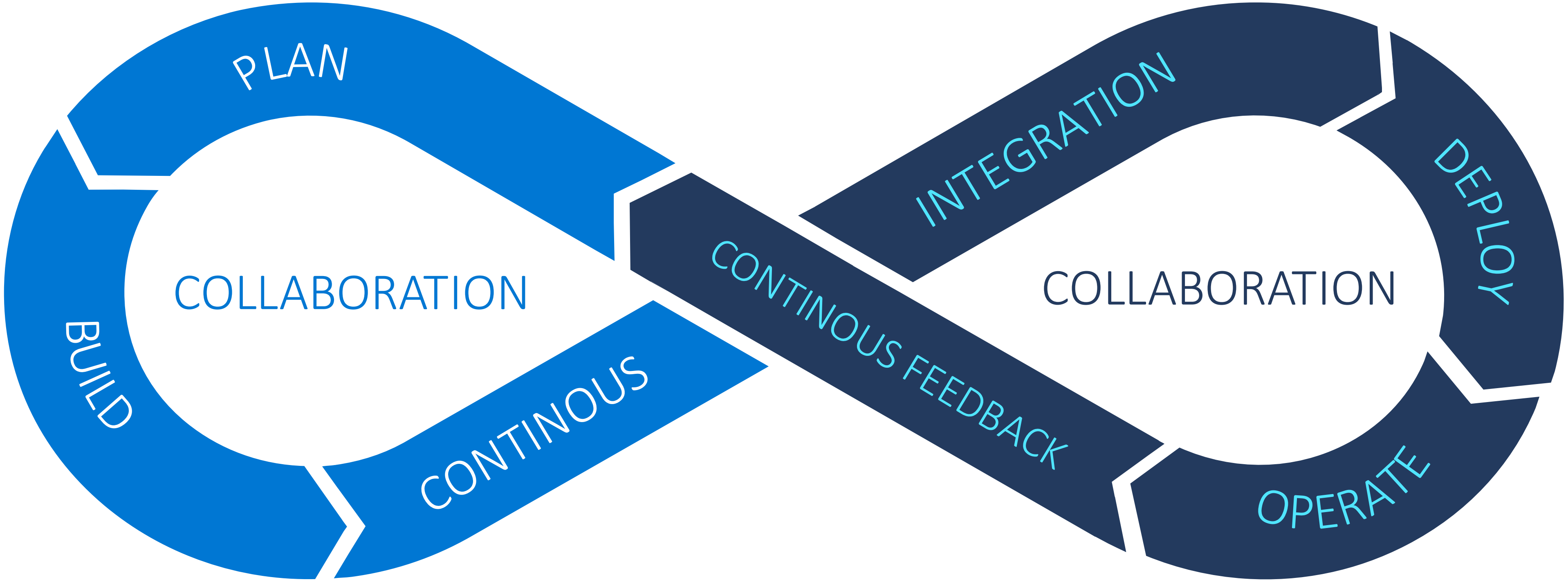
# Assume Breach: Mindset Change

- Mindset change is mandatory!
- Security is not only a ″network and firewall″
- Should not be played only by security team
- Security is responsibility and duty of everyone
- If applications are the main attack vector, we need to improve

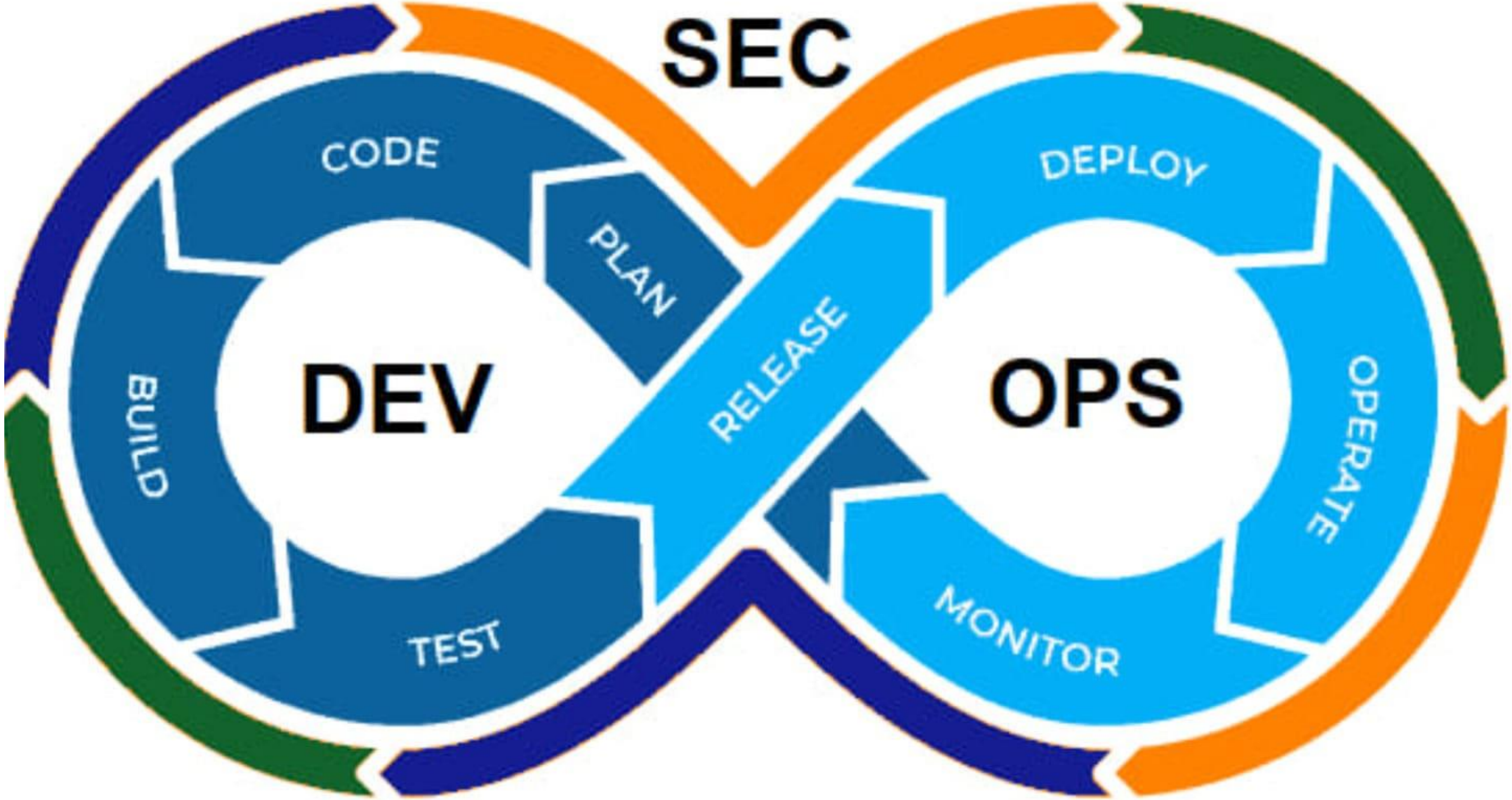
# Impact caused by a security breach

- Costs and efforts related with response and notification
- Lost employee productivity
- Lawsuits and settlements
- Regulatory fines and response
- Cost of fixing infrastructure
- Brand recovery costs & liabilities

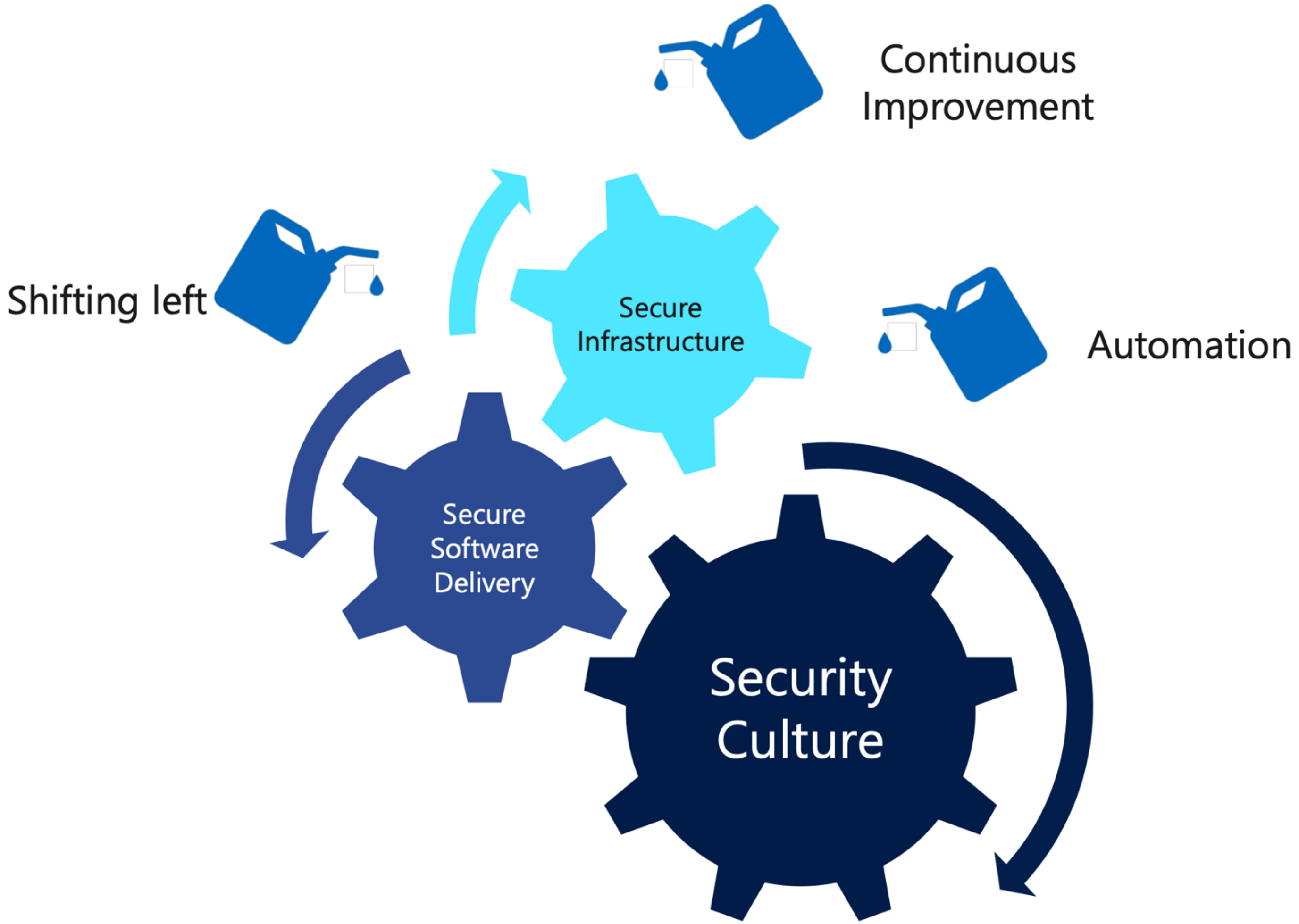
# DevOps: Infinite Loop



# DevSecOps: Infinite Loop



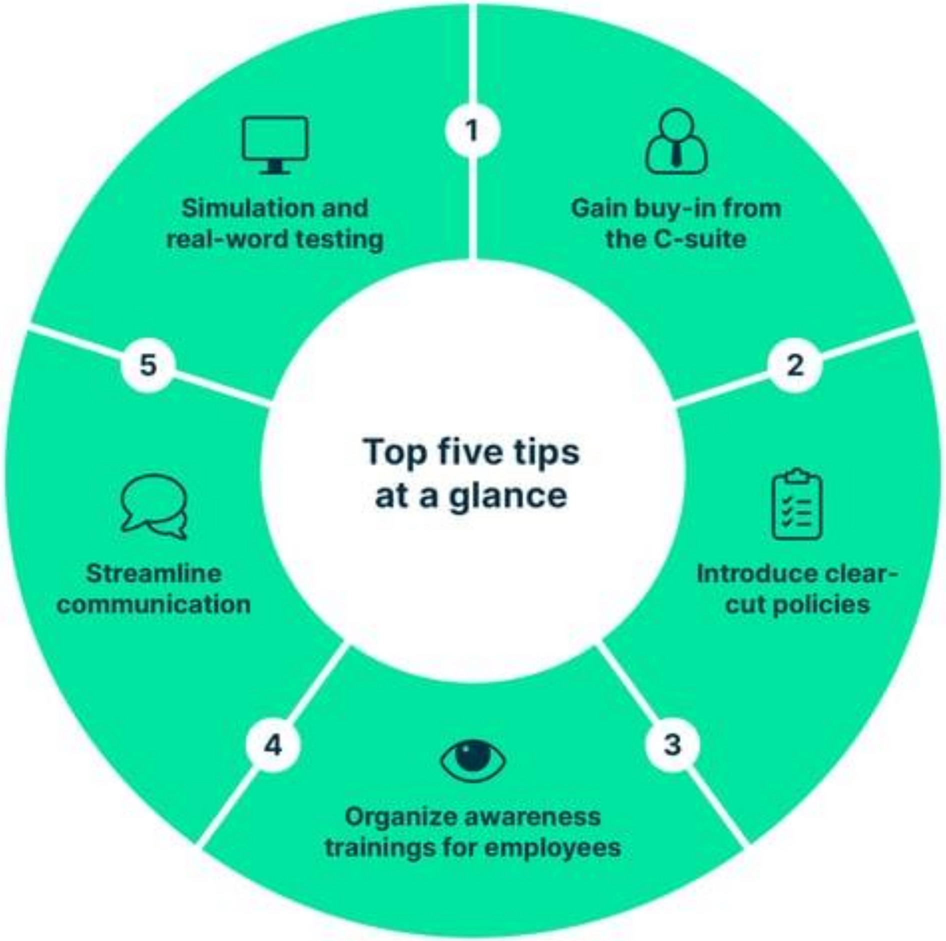
# Secure DevOps: Principles and Practices



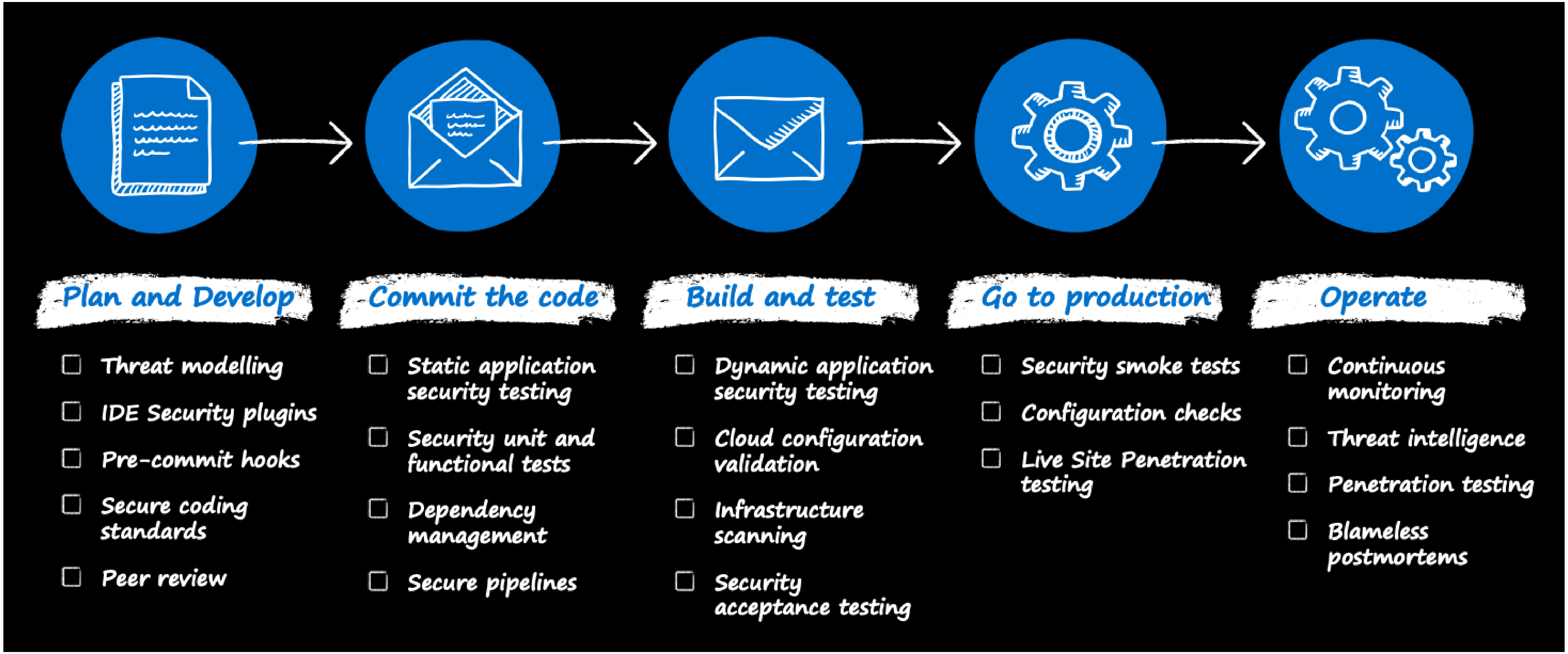
# Principles

Secure DevOps

# Build Security Culture


Top five tips at a glance
1
Gain buy-in from the C-suite
2
Introduce clear-cut policies
3
Organize awareness trainings for employees
4
Streamline communication
5
Simulation and real-word testing

# Secure Software Development Lifecycle



**Plan and Develop**

- □ Threat modelling
- □ IDE Security plugins
- □ Pre-commit hooks
- □ Secure coding standards
- □ Peer review

**Commit the code**

- □ Static application security testing
- □ Security unit and functional tests
- □ Dependency management
- □ Secure pipelines

**Build and test**

- □ Dynamic application security testing
- □ Cloud configuration validation
- □ Infrastructure scanning
- □ Security acceptance testing

**Go to production**

- □ Security smoke tests
- □ Configuration checks
- □ Live Site Penetration testing

**Operate**

- □ Continuous monitoring
- □ Threat intelligence
- □ Penetration testing
- □ Blameless postmortems

# What is DevOps Infra?

- For plan, your PO, designer or architect workstations are DevOps Infra
- On build phase, developers and any want producing code workstation is DevOps Infra
- If you use any internal repository, is DevOps Infra
- During CI/CD, your runners are DevOps Infra, even more if you don’t control them directly and you’re doing deploys on your infra
- On testing phases, testers and even customers workstations are DevOps Infra
- During operation, all your operations and infra team workstations are DevOps Infra
- Oh! And your production (all) environments are DevOps Infra too! ☺

# Secure DevOps Infra

- Vulnerable workstations open doors for lateral moves
- Constantly update your machines
- Zero trust principles, grant access to everything is needed but nothing more
- Repository access sharing credentials and adding to the repos
- Reuse of credentials without rotation
- Isolate your environments to make harder to do lateral moves
- Upskill your collaborators and make surprise tests for common tasks, like email phishing

# Main Practices

Secure DevOps

# Secure DevOps Practices

- Secure DevOps Practices acts as the enablers of principles
- Making these practices better allow you to implement better processes for your principles
- Makes security into your daily workflow
- Main practices
    - Shifting Left
    - Continuous improvement
    - Automation

# Shifting Left

- Introducing security controls since the beginning
- Security team must be involved since day 1
- Initially, can be to make solution compliant with well defined security controls
- Since security is an everyone’s responsibility, security teams can be focused on upskilling and being always updated

# Continuous Improvement

- Is a basic practice for DevOps, you must be always looking to your processes and try to make them better, faster, more secure
- Security topics are evolving every day, with attackers always one step forward than defenders
- You need to be informed by security team and your security controls, to identify possible vulnerabilities
- Your implementation processes needs to be reviewed constantly to face new possible vulnerabilities

# Automation

- Again, crucial practice to have a proper outcome
- Automate allow you to be consistent on analysis, faster on implementation and make it easier to evolve
- SCA and SAST processes without automation are not viable
- Can be used on every step and allow you to create security guards that only allow you to proceed when meet security principles

# Automation

- Again, crucial practice to have a proper outcome
- Automate allow you to be consistent on analysis, faster on implementation and make it easier to evolve
- SCA and SAST processes without automation are not viable
- Can be used on every step and allow you to create security guards that only allow you to proceed when meet security principles