What is DevOps?

DevOps is a term used to describe a set of cultural philosophies, practices and tools that bring together software development (Dev) and IT operations (Ops) and increase an organization's ability to deliver applications and services at high velocity.

With DevOps methods, new application functionality can be delivered frequently. Elastic cloud infrastructure accommodates surges in demand through auto-scaling processes that spin up new computing resources (virtual machines or containers) and deploy more instances of an application as required. Organizations pay only for the amount of computing needed.

Many enterprise organizations around the world are turning to agile and DevOps methodologies to eliminate functional and administrative silos and automate change management, configuration management and deployment processes. DevOps can ultimately help organizations accelerate time-to-market, improve product quality, eliminate inefficiencies, accelerate digital transformation and become more responsive to customer needs.

What is CI/CD?

By merging Development and Operations and driving more collaboration between the functions, DevOps enables shorter development cycles that are more closely aligned with business objectives. Using Continuous Integration (CI), developers merge code changes to a repository multiple times a day and the changes are automatically integrated into builds. The approach of Continuous Delivery (CD) requires that code always be in a deployable state so that it can be deployed to production at any time at the touch of a button.

DevOps Security Challenges

Despite its many benefits, DevOps presents new risks and cultural changes that create security challenges that cannot typically be addressed by conventional security management solutions and practices. These traditional approaches are often too slow, costly or complex to support automated software delivery and deployment into the cloud or as a container. These challenges include:

Privileged Credentials Used in DevOps Are Targeted by Cyber Attackers. One of the biggest security challenges in DevOps environments is privileged access management. DevOps processes require the use of human and machine privileged credentials that are very powerful and highly susceptible to cyber attacks.

- Human access: With high-velocity processes, DevOps practitioners require privileged access across development and production environments.
- Machine access: With automated processes, machines and tools require elevated privileges (or permissions) to access resources with no human involvement. Examples include:
  - Automation tools: Ansible, Puppet and Chef
  - CI/CD tools: Jenkins, Azure DevOps and Bamboo
  - Container management tools: Docker and Linux Containers (LXC)
  - Container orchestration tools: Kubernetes, Red Hat OpenShift, Pivotal, Cloud Foundry

Tier Zero assets, such as Ansible and Jenkins, have access to creden recognize this, and increasingly seek out privileged credentials including passwords, access keys, SSH keys and tokens, as well as other types of secrets such as certificates, encryption keys and API keys. Attackers can exploit unsecured credentials in DevOps environments, resulting in cryptojacking, data breaches and destruction of intellectual property.

Developers Are Focused on Velocity—Not Security. Focused on producing code faster, DevOps teams often adopt insecure practices outside of the purview of security teams. Such practices can include leaving embedded secrets and credentials in applications and configuration files, reusing third-party code without sufficient scrutiny, adopting new tools without evaluating them for potential security issues and insufficiently protecting DevOps tools and infrastructure.

Tool-centric Approaches to Secrets Management Create Security Gaps. DevOps tools often have some built-in features for protecting secrets. However, these features don't facilitate interoperability or securely sharing secrets across tools, clouds and platforms. Often, DevOps teams the built-in features of their individual tools to manage secrets. This approach can make it difficult to adequately protect the secrets, since they cannot be monitored and managed in a consistent manner.

STAY IN TOUCH

Steps for Enabling DevOps Security In Your Organization

Following are steps organizations often take to achieve DevOps security at scale while addressing the risks of privileged access and aligning to DevOps culture and methods:

- **Instantiate security policy as a code.** A cornerstone of DevOps is the concept of "Infrastructure as Code" (sometimes referred to as immutable infrastructure), which supplants the traditional model of manually administering and configuring servers and software. By applying this concept to security—instantiating and managing security policy as code—organizations can eliminate manually intensive, error-prone configuration processes.
- **Establish separation of duties.** Distinct roles and responsibilities should be clearly defined within a DevOps team:
  - Developers should focus on creating applications to drive business results.
  - Operations should focus on delivering reliable and scalable infrastructure.
  - Security should focus on safeguarding assets and data and mitigating risks.

Interactions between each group can be codified in a written security policy. For example, developers create the security policy that declares what privileges their application or service requires. Security staff then review and approve the security policy and operators make sure the application's deployment goes as expected.

- **Integrate security into CI/CD practices.** Too often in DevOps, security is treated as an afterthought and performed too late in the process, if at all. Then the potentially substantial last-minute changes needed to address vulnerabilities result in delayed releases. Forward-looking organizations are using advanced workflow scheduling and management tools like Kanban to model flows, accelerate development and eliminate inefficiencies. Additionally, security teams are increasingly deconstructing applications into microservices to simplify security reviews and changes.
- **Take a proactive approach to security.** Strong security practices should be instituted throughout the application lifecycle to reduce vulnerabilities, improve security posture and mitigate risks. Good DevOps security hygiene practices include:
  - Address security requirements and potential vulnerabilities holistically, since attackers may only need to exploit one vulnerability to carry out their mission.
  - Reduce the concentration of privilege in build automation tools and ensure that code repositories do not expose secrets.
  - Maintain secrets used by machines and people (passwords, cert

- Establish a baseline for normal usage patterns to detect anomalies so that malicious users are traceable and cannot steal credentials.
- Instill accountability by recording how credentials are used. For example, for human users, consider keystroke or video logging of the session.
- Provide each machine its own unique identity in order to audit and monitor its access to secrets.
- Run vulnerability scans and conduct penetration tests to improve cybersecurity posture.
- Educate developers on security threats and best practices.
- Foster close cooperation and collaboration between security and development teams.

- **Automate security processes.** DevOps uses automation to accelerate application lifecycle management and remove human latency. Similarly, DevOps security should leverage automation to minimize human interaction and manual intervention. For example, by automatically rotating secrets—passwords, keys, certificates—organizations can prevent attackers from gaining access to DevOps tools, access keys or systems for an extended period of time. Automated security procedures can also be used reactively if a breach is detected. For example, privileged sessions can be automatically terminated and credentials automatically rotated the moment a security breach is identified.

Learn More About DevOps Security

- CISO View: Protecting Privileged Access in DevOps and Cloud Environments
- CyberArk Application Access Manager Datasheet
- CyberArk Solutions: DevOps Security
- How CISOs At Leading Global Organizations Secure Their DevOps Environments
- Securing DevOps Environments In The Enterprise With CyberArk Application Access Manager

STAY IN TOUCH

OTHER GLOSSARY ENTRIES

A

Active Directory

Adaptive Multi-factor Authentication

App Gateway

C

Cloud Security

D

Data Breach

DevOps Security

E

Endpoint Security

I

Identity and Access Management (IAM)

Identity as a Service (IDaaS)

J

Just-In-Time Access

L

M

Passwordless Authentication

Privileged Access Management (PAM)

Ransomware

Robotic Process Automation (RPA)

S

SaaS

Secrets Management

Security Assertion Markup Language (SAML)

Single Sign-On (SSO)

V

Virtual Directory

Z

Zero Trust

STAY IN TOUCH

PRODUCTS

CORE PRIVILEGED ACCESS SECURITY

CYBERARK IDAPTIVE

CLOUD ENTITLEMENTS MANAGER

ENDPOINT PRIVILEGE MANAGER

CYBERARK ALERO

CYBERARK PRIVILEGE CLOUD

APPLICATION ACCESS MANAGER

SOLUTIONS

SOLUTIONS OVERVIEW

AUDIT AND COMPLIANCE

SECURITY AND RISK MANAGEMENT

INDUSTRY SOLUTIONS

COMPANY

WHY CYBERARK

COMPANY OVERVIEW

CAREERS

INVESTOR RELATIONS

BOARD OF DIRECTORS

MANAGEMENT TEAM

NEWSROOM

OFFICE LOCATIONS

PATENTS

CORPORATE RESPONSIBILITY

SERVICES AND SUPPORT

SECURITY SERVICES

RED TEAM SERVICES

TRAINING & CERTIFICATION

TECHNICAL SUPPORT

EPM SAAS REGISTER / LOGIN

PRODUCT SECURITY

PRODUCT DOCUMENTATION

CONTACT

REQUEST A DEMO

SCAN YOUR NETWORK

FOLLOW US

Hey there 👋Welcome to CyberArk! What led you to stop by today?

1