
# Azure DevOps Security CheckList

*Version:2.0*

06.07.2024

Okan YILDIZ
Secure Debug Limited
Senior Security Engineer / Senior Software Developer
| CASE .NET | CEH | CTIA | ECIH | CCISO |

# Summary

This Azure DevOps Security Guide, prepared for Secure Debug Limited, provides a comprehensive framework for ensuring a secure and compliant Azure DevOps environment. The guide covers various aspects of security, including access control, network security, code security, and continuous monitoring.
Key points addressed in this guide include:

1. Managing users and groups using Role-Based Access Control (RBAC) to define and enforce granular permissions.
2. Applying the principle of least privilege for granting permissions to minimize potential risks.
3. Regularly reviewing user accounts and disabling unnecessary accounts to reduce the attack surface.
4. Implementing strong authentication with Multi-Factor Authentication (MFA) to protect against unauthorized access.
5. Integrating centralized identity management using Single Sign-On (SSO) and Azure Active Directory.
6. Reducing authentication risks using risk-based policies and Azure AD Identity Protection integration.
7. Restricting access with IP-based network security groups and private networks.
8. Establishing secure communication with on-premises systems using VPN or ExpressRoute.
9. Protecting and routing network traffic with Azure DDoS Protection and Azure Firewall.
10. Applying code review processes and utilizing static and dynamic code analysis tools for vulnerability detection.
11. Establishing secure coding standards and ensuring dependency security.
12. Incorporating security controls and automated tests in Build and Release pipelines.
13. Securing agents with trusted agent pools and implementing Git branch policies and pull request reviews for code security.
14. Storing credentials, certificates, and access keys securely in Azure Key Vault and configuring access for Azure DevOps pipelines.
15. Monitoring changes using Azure DevOps audit logs for security, compliance, and operational awareness.
16. Continuously tracking and improving security posture with Azure Policy and Azure Security Center.
17. Conducting internal and external security audits and penetration tests for evaluation and continuous improvement.
18. Regularly review and update the security configurations of your Azure DevOps services, resources, and tools.
19. Implement secure baselines for your Azure resources and enforce them consistently across your environment.
20. Use Azure Policy to define and enforce security configurations across your Azure resources.
21. Continuously monitor configuration changes and assess their impact on your security posture.

22. Implement a robust backup and recovery strategy for your critical data, including source code, artifacts, and configuration data.
23. Use Azure Backup and Azure Site Recovery to protect your data and applications.
24. Regularly test your data recovery processes to ensure they are effective and up to date.
25. Establish a disaster recovery plan to minimize downtime and data loss in case of a security breach or system failure.
26. Maintain an up-to-date inventory of all Azure DevOps resources, including repositories, pipelines, environments, and tools.
27. Use Azure Resource Manager (ARM) templates to manage your Azure resources in a consistent and automated manner.
28. Implement tagging strategies to categorize your Azure resources based on project, team, or other relevant attributes.
29. Continuously monitor your inventory and resources for any unauthorized changes or access.
30. Securing containers by using best practices for container security in Azure, including the use of official and verified images, implementing the principle of least privilege, and setting resource limits.
31. Using Azure Kubernetes Service (AKS) securely by configuring cluster and network security, enabling role-based access control, and regularly updating and patching AKS nodes.
32. Conducting regular vulnerability scanning for containers using integrated tools in CI/CD pipelines and runtime scanning to detect and remediate security risks.
33. Implementing continuous security monitoring using tools like Azure Monitor, Azure Security Center, and Azure Sentinel to detect and respond to potential threats in real-time.
34. Setting up alerts and notifications to promptly respond to security incidents and integrating monitoring with Azure DevOps for seamless security oversight of DevOps processes and applications.

This summary highlights the main topics covered in the guide, providing a holistic approach to Azure DevOps security, aimed at fostering a culture of continuous improvement and collaboration between developers, security teams, and other stakeholders. Implementing these best practices will contribute to the ongoing success of your DevOps projects and help protect your organization's critical assets.