Microsoft Security

Security & Privacy Considerations in App Development

Presented by:
Richard Antonow
Security Technical Specialist - Microsoft

Agenda

- The threats
- Zero Trust – the first step
- Design and develop secure code
- Protect your work
- Defender for Dev Ops
- What is this AI thing?

Lets start with some definitions…

CODE: Noun: *Program instructions used to develop an application*

DEBUG: Noun: *The process of identifying and taking bugs OUT of computer software*

PROGRAMMING: Noun: *The process of putting bugs IN computer software*

PROGRAMMER: Noun: *Developer that runs primarily on Mountain Dew and Pizza*

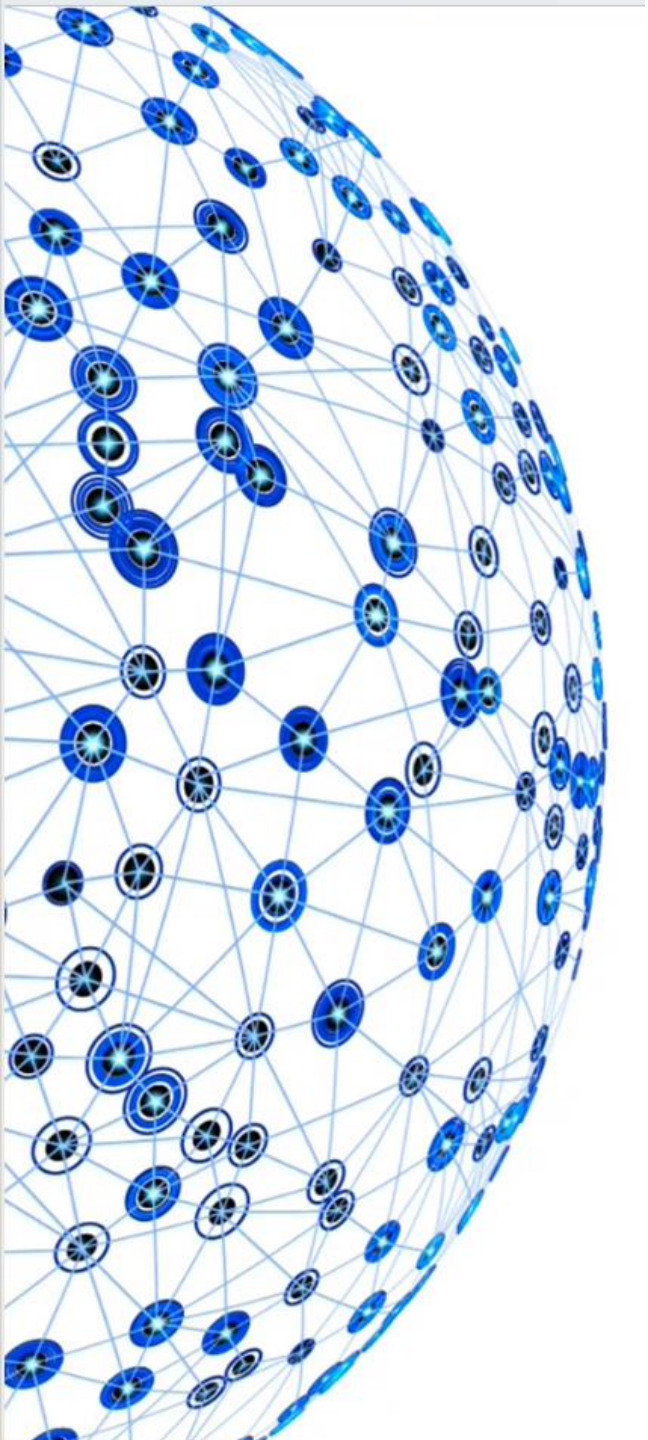
THREAT: Noun: *An action triggered by a threat source to exploit a specific vulnerability*

BAD ACTOR: Verb: *A person or organization that facilitates security threat events* …also see “Adam Sandler”

Bad Actors will continue to ramp up threats against government, education, retail and public utilities. AI will only make these attacks more prevalent and successful

In 2023, Microsoft invested $13B in OpenAI technology (being incorporated as “CoPilot)

According to FBI's Internet Crime Complaint Center (IC3),



Current software develop trends, low-code/no-code, Software – as-a-Service (SaaS), multi-cloud Single-Sign-On (SSO) gives threat and nation state actors more targets for attack.

Man-in-the-middle Attacks

Phishing Attacks

Brute Force Attacks

Malware infections

Adversary-in-the-Middle (AiTM) Attacks

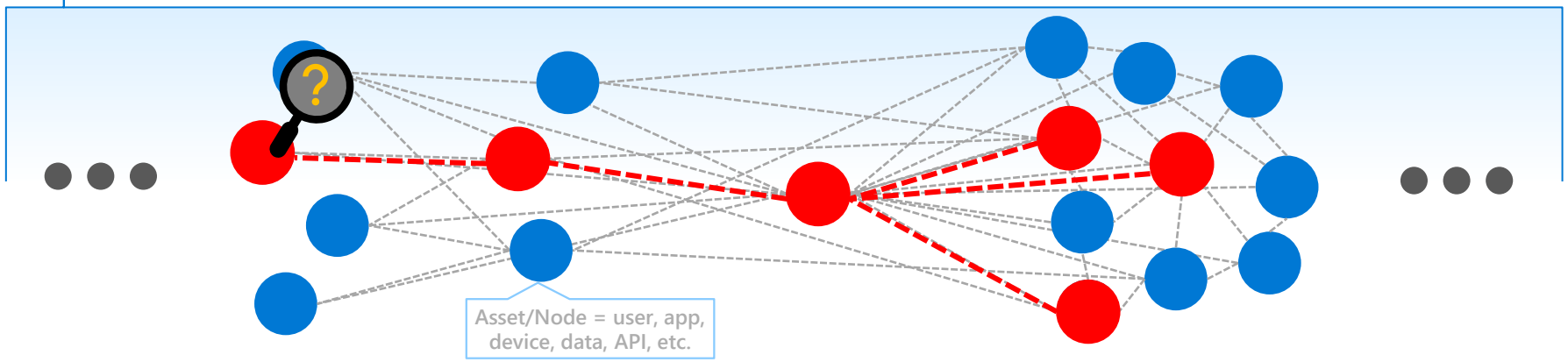
[Keep Up With the Latest Security Trends and Threats in Software Development (devprojournal.com)]

Designing Secure Code

Zero Trust Principles

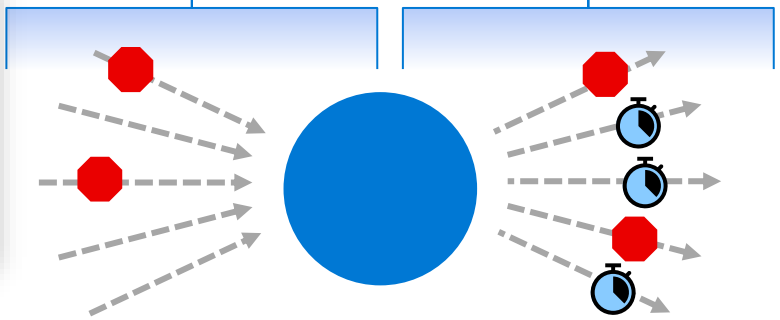
Assume Breach (Assume Compromise)
Assume attackers can and will successfully attack anything (identity, network, device, app, infrastructure, etc.) and plan accordingly

→ *Transforms overall thinking, strategy, and architectures from "safe network" to "open network"*



Verify explicitly
Protect assets against attacker control by explicitly validating that all trust and security decisions use all relevant available information and telemetry.

→ *Reduces "attack surface" of each asset*



Use least privilege access
Limit access of a potentially compromised asset, typically with just-in-time and just-enough-access (JIT/JEA) and risk-based polices like adaptive access control.

→ *Reduce "blast radius" of compromises*

Secure Code Design Principles

- The Microsoft Security Design model

Microsoft Security Development Lifecycle (SDL)

SDL is a software development process from Microsoft that helps developers build more secure software.

Training – Ensure App developers are properly trained to develop safe and secure code

Requirements – Define and continually update security requirements for the applications

Design – Establish standard security features and protocol standards all coders should use

Implementation – Ensure applications and supporting infrastructure are proper for the design

Verification – Leverage independent test teams*, UAT testing, security PEN testing

Release – Stage release of applications through pre-release / preview / final release

Response – Establish a standard Incident Response (IR) process and plan to address threats

*Static Analysis Security Testing (SAST), Dynamic Analysis Security Testing (DAST)

Microsoft Security Development Lifecycle Practices

Design Secure Apps

- The initial steps set the precedence for all that follows

Training involves, not only understanding how to write code well but, learning WHAT the threats are and how attacks work

Gathering **requirements** is one of the MOST crucial steps in proper code development. This part is your opportunity to address crucial success factors:

- Understand risks associated with security issues.
- Identify and fix security bugs during development.
- Apply established levels of security and privacy throughout the entire project.

During the **design** phase, you will come up with a proper configuration for your application that meets the identified requirements, provides security for sensitive data, keeps the attack surface low and provides access to alerts and activity logging

Design secure applications on Microsoft Azure | Microsoft Learn

Develop Secure Apps

- Infrastructure and validation

Implementation is used to establish best practices for the use of the application, ensure the infrastructure is appropriate for supporting the application and prevention of security related issues. Azure Marketplace can provide you with tools to assist with the development:

SonarQube on Ubuntu 22.04 LTS

By AskforCloud LLC

SonarQube is an open-source tool that assists in code quality analysis and reporting.

GitLab CE Community Edition: Collaborate and...

By Cognosys

GitLab CE Community Edition is a powerful tool for efficient code management and collaboration on Oracle en...

DevOps All-in-one Platform

By VMLAB INC.

Pre-configured, customizable, secure, one-click to deploy GitLab Community Edition on Azure

The **verification** phase allows you to ensure the code meets the requirements defined earlier in the process for security, privacy and data protection. Verification also allows for validation of output data to ensure expected content

[Develop secure applications on Microsoft Azure | Microsoft Learn]

Deploy Secure Apps

- Deploying your code and on-going support

This phase is the cumulation of the overall process of secure code design and delivery.

The **release** phase is where the code is readied for public release. This phase may incorporate multiple stages from pre-release, preview to public release. Some key pieces include:

- Create an incident response plan
- Perform final security evaluation and processes
- Certify release of code

The **response** phase is "post release" and is a critical piece of the ongoing lifecycle of the application. During this phase, the application is monitored and processes put in place to watch for direct threats and Indicators of Compromise (IOC). Typically tools such as *Microsoft Defender for Cloud* are involved in this phase.

Deploy secure applications on Microsoft Azure | Microsoft Learn

Tips and Techniques

- Here are some general tips for creating safe and efficient code

- ✓ Don’t hard-code login credentials.
- ✓ Use user authentication to prevent brute force attacks.
- ✓ Randomize your session IDs.
- ✓ Don’t trust user input.
- ✓ Limit what your error codes say.
- ✓ Use automated tools.
- ✓ Always try to use strict mode in weakly typed languages like JavaScript.
- ✓ Validate data or files from the user by length and filetype.
- ✓ Use appropriate headers in the response that make sure to only allow data that is desired.
- ✓ Use secure protocols for communication between the application and any data resources / clients.

Secure coding guidelines for .NET - .NET | Microsoft Learn

Remember - when designing and writing your code, you need to protect the data being access by the application and, limit the access that code has to other resources to reduce the exposure

Why didn’t I mention Multi-factor Authentication?

How does Low/No-code differ from AI generated?

- **What is the difference between low-code, no-code and Generative AI code development?**

Low-code is achieved using tools that allow developers to create applications through a visual approach and "drag and drop" interfaces where the application then creates the code behind the scenes. Customization of the code to tailor it to specific needs may still be needed so some coding experience is required. Typically requires less training than traditional coding techniques.

No-code is also achieved using visual tools and "drag and drop" interfaces but it is heavily dependent upon canned templates and libraries and does not easily lead itself to customization. Has limited capabilities. Requires no formal coding experience.

Generative AI for code development doesn't use templates and libraries of components. It has access to massive compute power and large language models and data. To create applications, the developer writes, in "natural language," the plan for the application and the Generative AI solution suggests code snippets from scratch that will produce the desired results. This capability is still in its infancy and the created code should be cross-checked to ensure all the required parameters are met and that security and privacy are adhered to.

Microsoft is heavily invested in this capability and is deploying "CoPilot" capabilities for; MS Edge, Microsoft 365, Microsoft Sentinel and components

AI code-generation software: What it is and how it works - IBM Blog

Securing your work

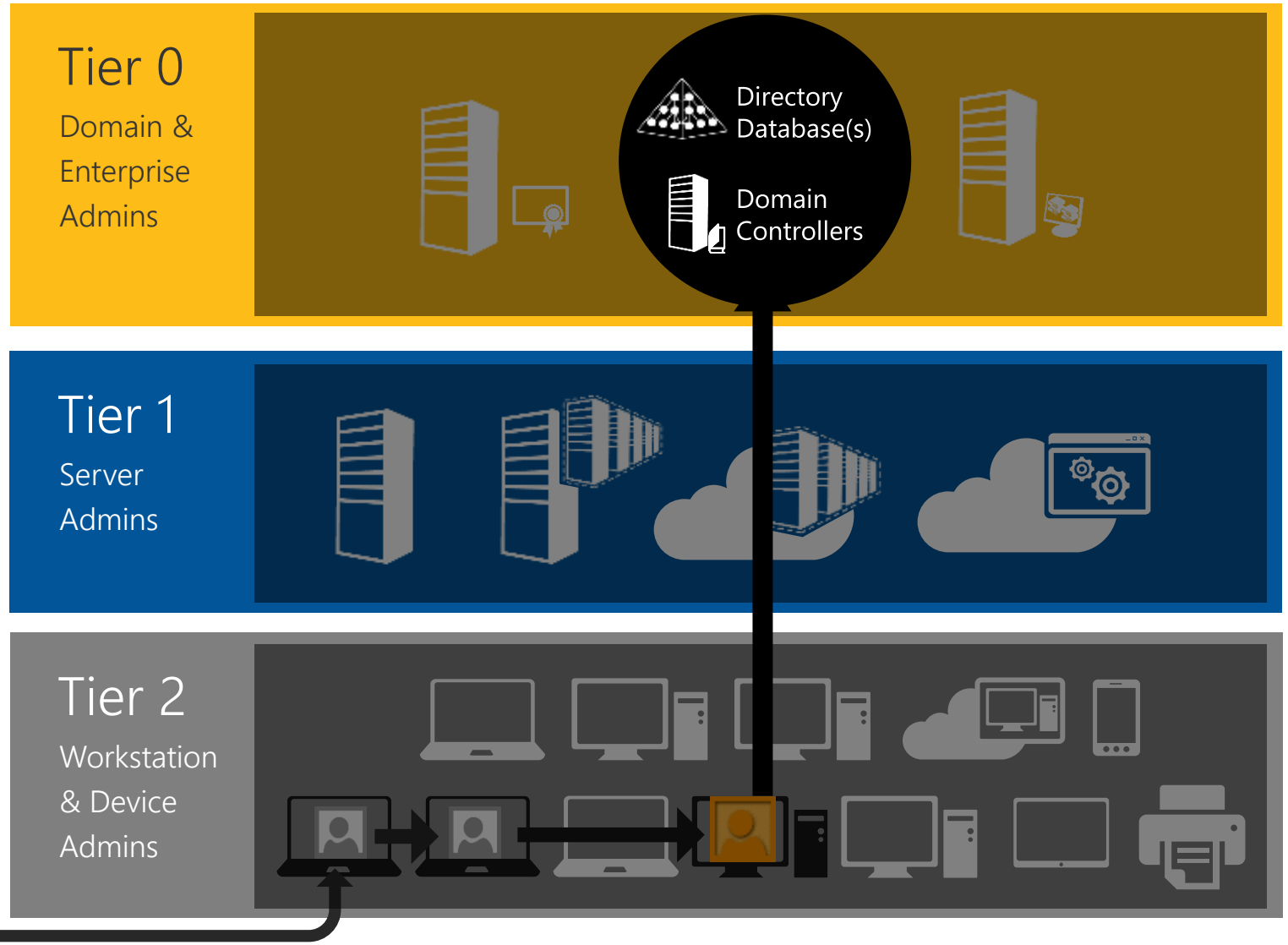
How does a typical attack happen?

What is the bad actor going after? Your company's intellectual property and data!!!

24-48 Hours

The steps to infiltration:

1. Establish an entry point (Phishing Attack, etc.)
2. Lateral Movement
   a. Steal Credentials
   b. Compromise devices / more credentials
3. Privilege Escalation
   a. Get Domain Admin credentials
4. Execute Attacker Mission
   a. Steal data, destroy systems, etc.
   b. Persist Presence



Securing Source Code

- You've written your code – now what?

Protection of your organizations data and source code is one of the **highest priorities**! Remember – "phishing" attacks are not going after users identities… they are going after the DATA!!!

Where you store your source code during development and, after deployment is part of the overall security plan and appropriate protections must be put in place including:

- Create a source code protection policy
- Prevent the use of insecure source code
- Implement access controls
- Use encryption and monitoring
- Deploy network security tools
- Don't forget about endpoint security
- Pay attention to patents & copyright
- Implement secure development practices

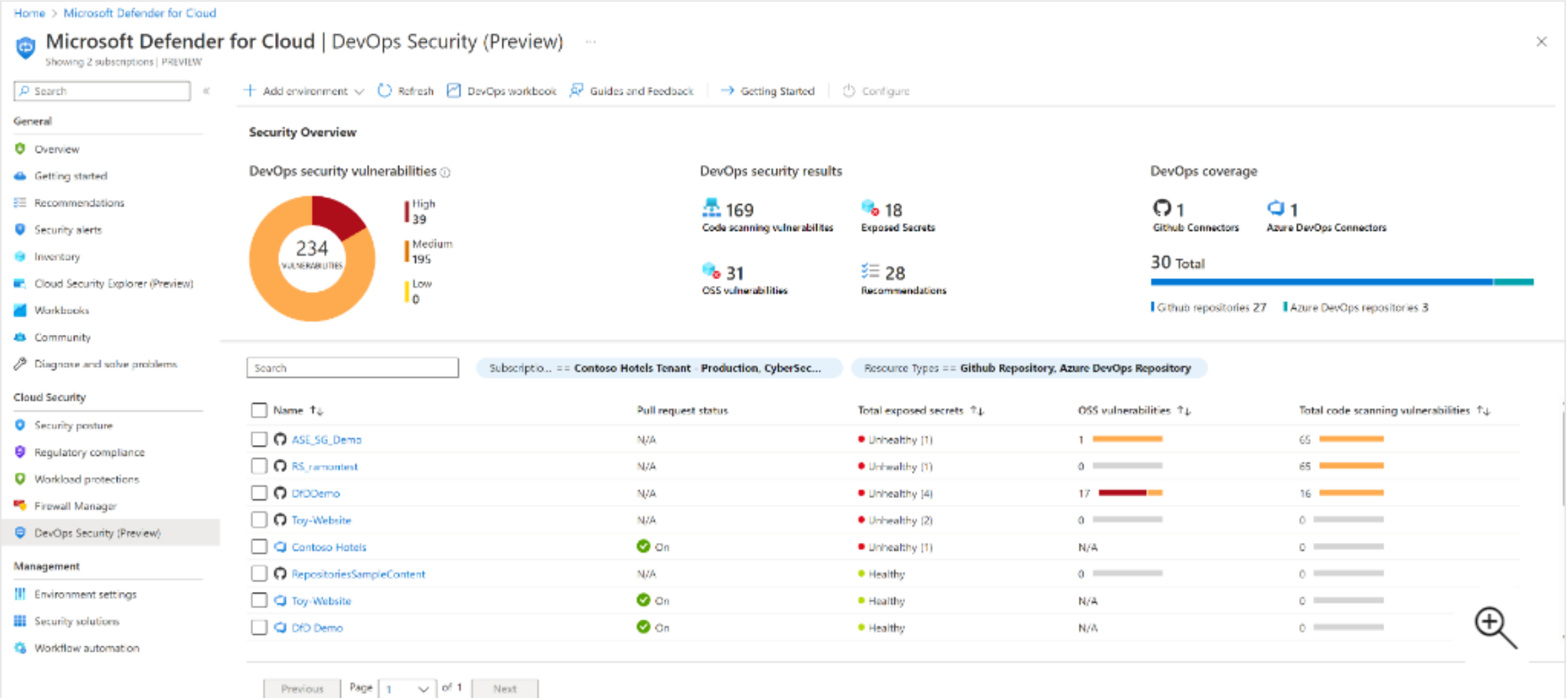
Securing your repository - GitHub Docs

Microsoft 365

Microsoft Defender for Dev Ops

Microsoft Defender for Dev Ops

Comprehensive visibility, posture management, and threat protection for Dev workloads

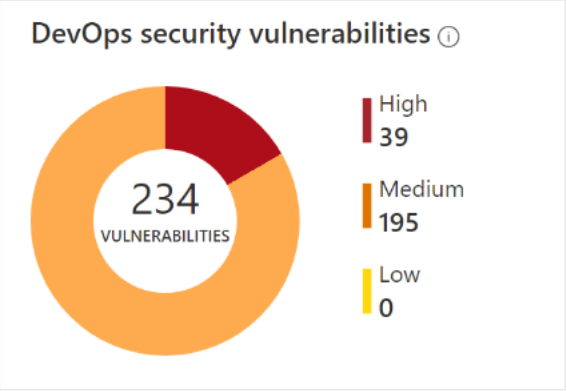
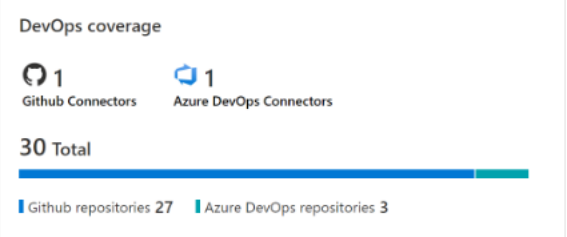


Microsoft Defender for Dev Ops

Discover vulnerabilities and security holes

What can the toolset do?

- Protect applications from code to cloud
- Correlate information with other cloud security solutions to provide contextual insights and prioritize remediation
- Provide inventory of all resources and external data connections being used by the code
- Identify exposed secrets in repositories
- Identify Open Source vulnerabilities
- Identify Infrastructure as code misconfigurations
- Perform code scanning for vulnerabilities and misconfigurations

Page section	Description
	Shows the total number of vulnerabilities found by Defender for DevOps. You can organize the results by severity level.
	Presents the total number of findings by scan type and the associated recommendations for any onboarded resources. Selecting a result takes you to corresponding recommendations.
	Provides visibility into the number of connectors and repositories that have been onboarded by an environment.

Microsoft Security Copilot

The first generative AI security product to help defend organizations at machine speed and scale

Microsoft Security Copilot

Defending at machine speed

"It takes us three minutes to do a task that used to take at least a few hours"

- Private preview customer

Enable **response in minutes,** not hours

Simplify the complex with natural language prompts and easy reporting

Catch what others miss with deeper understanding of your enterprise

Upskill your security talent with cyber-trained generative AI

Tell me about my latest incidents.

Are any of my machines affected?

Summarize this incident in PowerPoint.

Which alerts are being triggered the most?

What is log4shell?

How can I improve my security posture?

How can I improve my security posture

Jan 10, 12:38 PM

Here are three evidence-based recommendations to improve your security posture:

1. **Use Microsoft Intune ASR rules to prevent common attack techniques:** This will help protect your devices from common attack techniques. Currently, 438,174 out of 758,659 devices are impacted. Learn more about ASR rules here.
2. **Use Microsoft Intune to enforce the latest Defender security baseline:** This will ensure that your devices are up-to-date with the latest security settings. Currently, 438,174 out of 758,659 devices are impacted. Learn more about the Defender security baseline here.
3. **Use Microsoft Intune Endpoint Privilege Management to reduce exposure by enabling standard users and limiting local admin accounts:** This will help minimize the attack surface by reducing the number of local admin accounts. Currently, 438,174 out of 758,659 devices are impacted. Learn more about Endpoint Privilege Management here.

Sources

Microsoft Intune

Confirm Off-target Report Pin

Let’s see it!

Thank You