Microsoft Security

Building a comprehensive API security strategy

An integrated approach to API management and security

Chapter 1:

Introduction

Introduction

Application programming interfaces (APIs) are the backbone of modern applications. They power many of your core business functions and allow you to share data with your customers, partners, and developers, across applications and services.

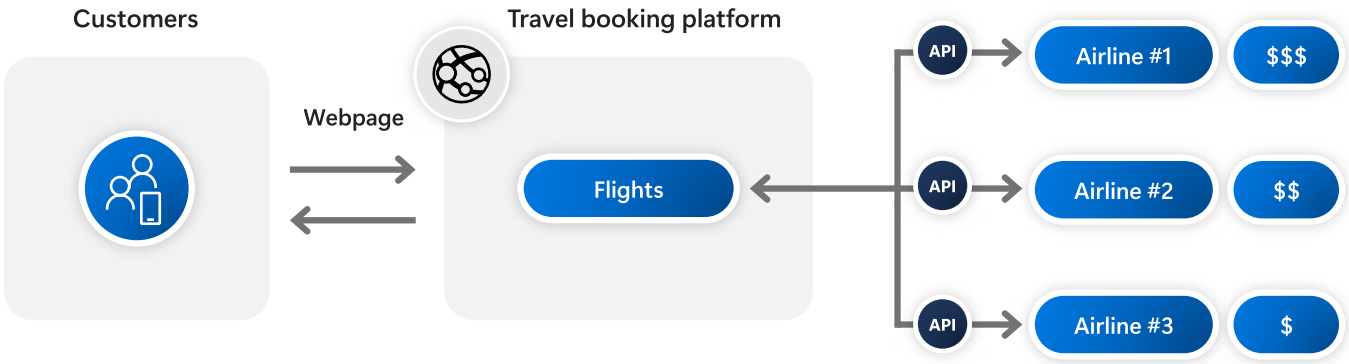
Fundamentally, an API defines how software components and systems should interact with each other. In recent years, APIs that send and receive requests over the web have become the standard way for modern applications and services to communicate.

To illustrate how common APIs are in the modern world, take the example of a simple interaction with a flight booking website:

1. A customer goes to the site and submits their search criteria.
2. The site pulls time and price information from multiple different airlines.
3. The website collects this data and returns the relevant information to the customer.
4. The customer selects and books a flight.

Each step in this relatively simple process relies on at least one API call to transmit information between various systems.

Over the last several years, online services have become more sophisticated, and the need for APIs to communicate between systems continues to grow. As



Sign in to the travel booking portal and check the tickets for their vacation

Interacts with different airline providers via APIs and responds with travel options

your organization takes advantage of the combination of cloud, AI, and the Internet of Things (IoT), APIs are also increasingly ubiquitous and more critical to your core business processes.

However, if APIs aren't sufficiently secured, they can pose risks to your data, operations, and systems. APIs, by their nature, are accessible programmatically, allowing threat actors to quickly scale up automated probes and attacks. In addition, traditional rules-based detection methods, lacking a behavioral understanding of your API traffic, may not detect subtle, initially legitimate-looking attacks against your API business logic.

The practice of API security aims to mitigate these risks by protecting your APIs from unauthenticated or unauthorized access, data exfiltration, and logic abuse. It involves assessing and managing your security throughout the API lifecycle, from design and development to deployment and operations, all the way to retirement and decommissioning.

Understanding and improving your API security posture—the overall state of your organization's security strength, resilience, and preparedness to respond to threats pertaining to APIs—should be one of your main priorities. Maintaining continuous, real-time visibility into the health of your API assets is crucial. This enables you to identify and understand potential security risks, allowing you to implement proactive measures to secure your environment before any attacks can exploit vulnerabilities.

API discovery is the first step in establishing an API security strategy. This process involves gaining visibility and establishing governance and active management over all APIs that your organization uses. This includes documenting APIs that you already manage, finding currently unmanaged APIs that need integration into your management systems, identifying unknown or "shadow" APIs that lack documentation but are still used in your business processes, and identifying ghost or third-party APIs that are used by your applications but may have inherent unknown vulnerabilities.

Discovery is followed up by evaluating your API security posture. This includes assessing your APIs against security best practices, identifying vulnerabilities, and discovering insights to understand the security risk profile of your APIs. You use tools to help ensure that your APIs are securely implemented and that they won't expose your critical applications or sensitive business data to malicious actors. In addition, you need to develop a contextual understanding of how your APIs work and how the attack surfaces they present impact your application security more generally.

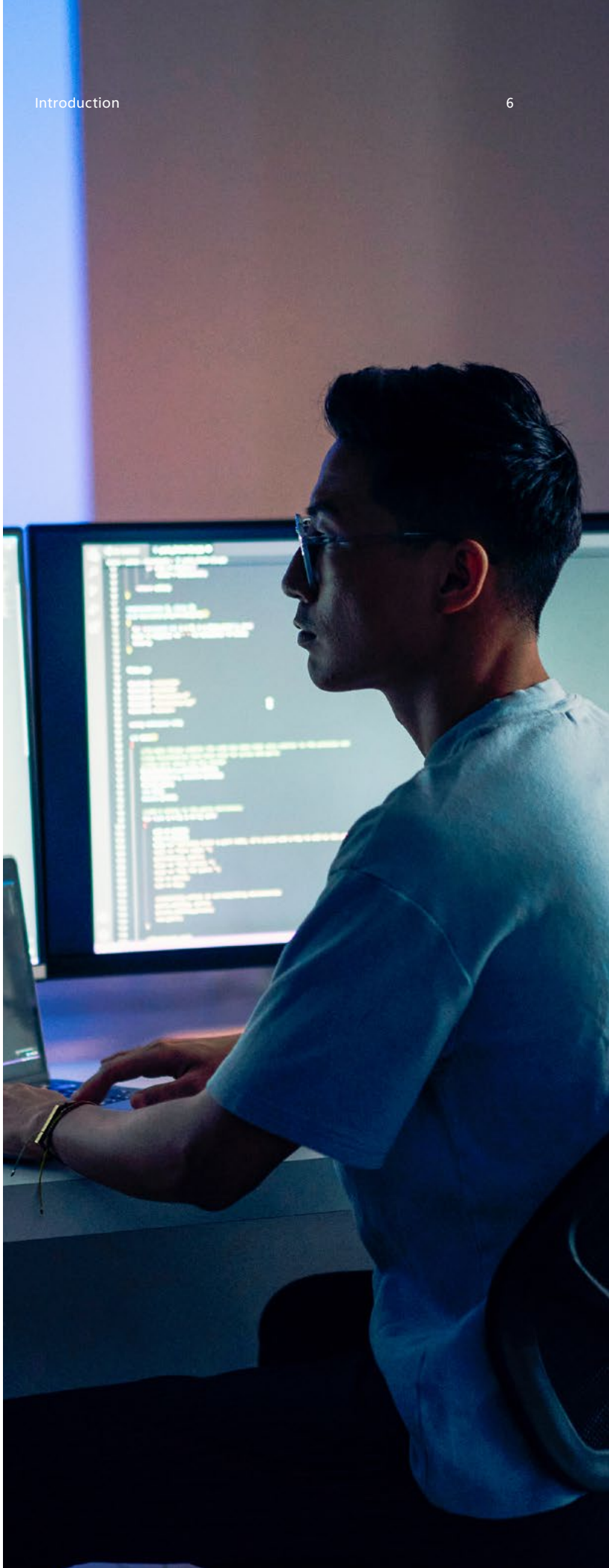
Finally, you need to monitor and protect your APIs against attacks in runtime. You should analyze API traffic for expected threats, but more importantly, you should examine usage

behavior to find unforeseen anomalies and threats. All of this is achieved by building a layered defense made up of complementary tools and systems.

In this e-book, you can explore how to approach API security as a comprehensive discipline to help you:

- Discover your APIs and their exposure level.
- Assess and improve your API security posture.
- Monitor and protect your APIs from threats and attacks.
- Build layered defenses for your APIs by using multiple complementary tools, services, and techniques.

In addition to covering these API security fundamentals, this e-book provide examples of the tools and services you can use to apply these concepts within the Microsoft ecosystem, for example Azure API Center, Azure API Management, Microsoft Defender for Cloud, and other complementary Microsoft security and management services.



Chapter 2:

API security protects your business-critical data and operations

API security protects your business-critical data and operations

APIs are everywhere. They allow your cloud-based, on-premises, and other connected applications to communicate with each other. They enable modern technological innovation by simplifying access to information and facilitating automation. And they allow businesses like yours to offer new products and services, create new revenue streams, and improve customer experience.

As your business continues to take advantage of the digital transformation, your reliance on APIs increases. The growth of modern cloud and mobile applications, the shift from monolithic to microservice architectures, and the growing importance of generative AI all are supported by an increasing use of APIs.

However, APIs handling potentially sensitive data are a tempting target for hackers and other malicious actors. Data breaches involving API vulnerabilities can cause significant damage.

Here are just a few recent examples of the damage that recent API breaches have caused:

- In 2023, a US-based telecommunications company disclosed an API data breach affecting 37 million customers. This event incurred costs related to customer notifications, credit monitoring services, potential regulatory fines, and class action lawsuits leading to losses estimated in the hundreds of millions of dollars.
- In early 2024, a leading global automotive manufacturer discovered an API breach caused by an insecure access token. Hackers were able to access sensitive blueprints, documents, source code, and passwords and access keys.
- In mid-2024, a major international computer retailer experienced an API-related data breach that led to hackers downloading 49 million customer records. Sensitive customer information, such as names, home addresses, and order information, were included.

The proliferation of APIs represents a rapidly increasing attack surface when compared to other parts of your infrastructure. In addition, unlike traditional web applications, APIs are designed for machine-to-machine

communication, introducing a unique set of potential vulnerabilities that your security, IT, and development teams need to manage.

2.1 Understand API threats

To help identify, understand, and deal with these threats, the Open Worldwide Application Security Project (OWASP), a nonprofit foundation that works to improve the security of software, maintains an API Security Top 10 list. You can reference this list to evaluate your APIs for common security weaknesses and to find ways to further harden your infrastructure.

With OWASP guidance in mind, consider the following set of questions as you start to think critically about your API security strategy:

- **Best practices, management, and governance standards for APIs.** Do you have an API inventory, and do you have an up-to-date view of this inventory? Do your development teams have API-specific security standards and best practices integrated into their DevOps workflows? What governance does your organization apply to your APIs? The answers to these questions will impact your ability to apply consistent security policies, enforce compliance, and detect potential threats.
- **API vulnerabilities and data protection.** Have you analyzed the data that your APIs handle, documenting sensitive information they could expose? Have you validated that only authenticated and authorized users and services can access your APIs and data? Are you encrypting all API traffic, and have you made sure that you have mechanisms in place to deal with volumetric attacks, like distributed denial-of-service (DDoS) attacks? Proactively identifying and mitigating vulnerabilities is key to improving your API security.
- **Monitoring and threat detection.** Are your monitoring systems able to record and analyze API requests and responses, errors and exceptions, performance, and availability? Can you analyze traffic and behavior patterns to detect unusual API usage? What capabilities do you have to help detect and respond to malicious and anomalous activities? The threat landscape is always changing, and security software that's able to monitor your APIs and automatically detect unauthorized access, data leakage, and attack attempts is critical to securing your APIs.

To protect your data, you need an organization-wide plan to secure your APIs. In the following sections, we discuss approaches for building this plan, including identifying your API risk surface and inventory, managing that inventory, and using a variety of security tools and technologies to build a combined defense strategy for your API security.

Chapter 3:

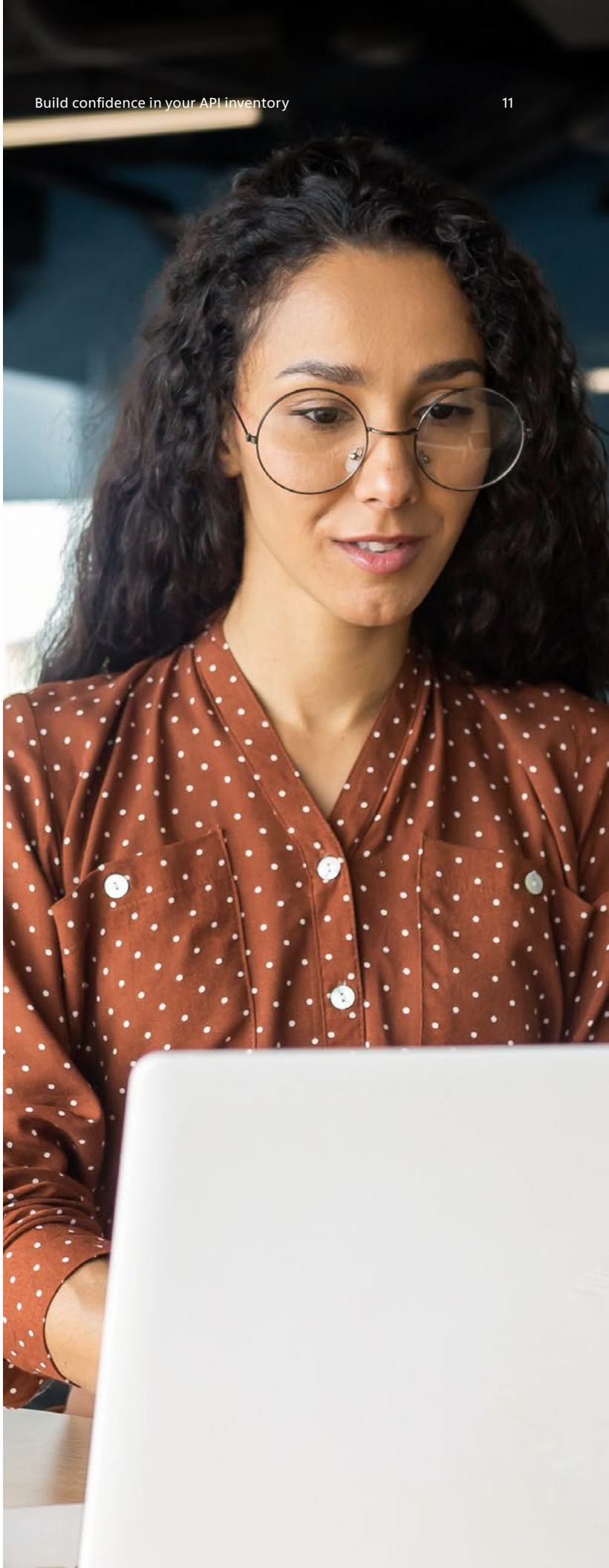
Build confidence in your API inventory

Build confidence in your API inventory

You can't protect what you don't know exists. Without an understanding of what APIs you host or use, you lack the ability to secure them against potential security risks, such as unauthorized access, data leaks, account takeover attempts, and DDoS attacks.

Many organizations lack a thorough API inventory, or they have a static inventory built using manual or ad hoc methods to discover and document APIs. This can lead to incomplete, inaccurate, and outdated information, resulting in undetected vulnerabilities.

OWASP risk API9:2023 Improper Inventory Management describes several example risk scenarios that can be caused by an inaccurate API inventory. For example, if your inventory doesn't cover staging or development API endpoints, networking or security fixes may not be applied to them. Alternatively, if documentation about an API is missing information about the types of data it handles, you may not even be aware that it exposes sensitive data.



3.1 Use API discovery to get a handle on your API inventory

The API discovery process aims to minimize these risks by developing a thorough understanding of your exposed API risk surface. It focuses on identifying, documenting, and understanding all APIs that your organization uses, resulting in a comprehensive and up-to-date inventory.

In the inventory phase of API discovery, you start by working with stakeholders across your IT and development teams to establish standards for documenting your APIs. Using these standards, you catalog and document known APIs, in addition to investigating undocumented "shadow" APIs that are used but not centrally managed or documented by your organization. You also create the ongoing change management processes to track when developers update or create APIs, helping to ensure that your API catalog and documentation are kept up to date.

After inventorying your APIs, you work with your development and security teams to identify risks that each of your APIs may pose to your organization's data and business processes. You need to look at all environments that host your APIs, including on-premises servers, shared datacenters, software as a service (SaaS) products, and cloud services. Your change management processes should also include updated risk evaluation as your APIs are modified.

Common questions you need to answer during the discovery process include:

- How many APIs does your organization have?
- Are these APIs being used?
- Who owns the APIs?
- Who consumes the APIs?
- Were they developed internally or provided by a third-party partner or a vendor?
- Where are the APIs hosted, and what is the process to govern them?
- Are the APIs used internally, or do they have external users?
- Which data or business process does each API interact with?

After discovery is complete, you should have a thorough and accurate picture of your organization's API inventory. You should also have identified any high-priority issues that need to be immediately addressed.

If you lack the required information to accurately inventory and catalog an API, assign relevant stakeholders to investigate the missing information. You can then either integrate the API into your governance and management systems or decide to deprecate it.

API DISCOVERY ON THE AZURE PLATFORM

As you work through this process, you can use Azure API Center to help you document your findings and prepare for the next steps in improving your API security posture.

Azure API Center allows you to create and maintain a centralized inventory of your organization's APIs, across types, lifecycle stages, and deployment location. In addition, you can document key information about your APIs, such as version details, API definition files, and common metadata.

Chapter 4:

Actively manage your dynamic API inventory

Actively manage your dynamic API inventory

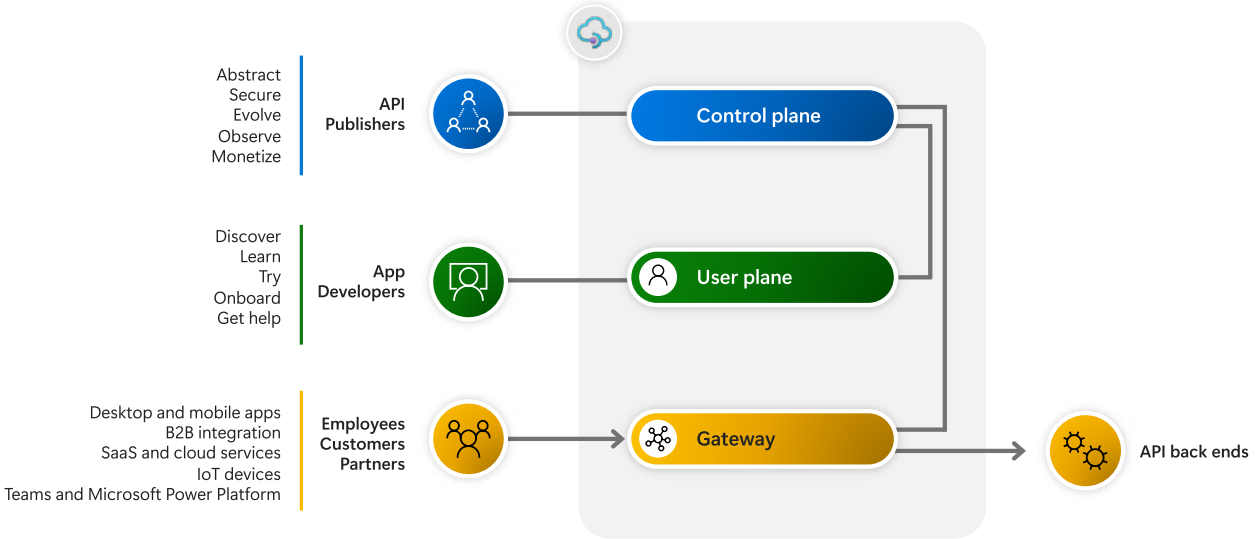
The API discovery process can give you a solid understanding of your current API inventory. However, after you've documented your APIs, you need to apply organizational policies and security controls, while dealing with any vulnerabilities you've identified.

In addition, your API inventory can change anytime your developers make an update to an application. An out-of-date API inventory could leave you unaware of new risks.

For example, failure to document API updates could prevent your security teams from correctly classifying newly sensitive data included in responses, potentially exposing that data. Alternatively, a newly created undocumented API could be missing a required security configuration, for instance DDoS mitigation, leaving it exposed to attack.

Maintaining an updated view of your API risk surface, while applying up-to-date mitigations for any identified threats, is key to keeping your APIs safe.

API management platforms are comprehensive sets of tools and software services designed to help you stay on top of your API inventory. They do this by streamlining and automating many of the design, deployment, security, and monitoring processes through the entire API lifecycle, from the start of design to when your API is decommissioned.



4.1 Use an API management platform

Integrating an API management platform into your IT processes is important when developing your API security plan and is a key step toward mitigating many of the OWASP-identified vulnerabilities, along with other threats. These platforms generally offer several common features to help you manage and secure your APIs:

- **API gateways.** Gateways are proxies that provide a single canonical endpoint for accessing your APIs. The gateway forwards requests to the correct back-end API services, enabling tasks like API traffic, routing, load balancing, and policy enforcement to be managed centrally.
- **Design and documentation capabilities.** These help your API developers design to code-quality and security standards, integrate testing mechanisms into your development pipelines, and make sure that your APIs have up-to-date documentation.
- **Centralized management capabilities.** The API management platform gives you a single control plane, allowing you to configure services, manage gateways, define API schemas, package and organize APIs, set up policies, review analytics, and manage users.
- **Authentication and authorization standards.** Your APIs should use standard methods, like OAuth, API keys, and JWT tokens, to make sure that only authorized users or systems can access them. API management platforms include capabilities to enable these mechanisms and enforce their use.
- **Developer portals.** A developer portal is a website that provides documentation, support information, and management functionality used by API developers and consumers.
- **Monitoring and analytics.** API management platforms include monitoring and analytics capabilities that can integrate with your wider IT alert and reporting systems.
- **Lifecycle and version management.** Lifecycle and version management oversees APIs from deployment and maintenance to deprecation and retirement, helping to ensure consistent, secure, and efficient performance throughout their lifespan.

4.2 Streamline API management with API gateways

API gateways are a particularly important component for managing and securing your APIs. By serving as a proxy between client requests and your actual back-end APIs, they centralize the flow of API requests and provide a single point of management. You can apply a central set of API management governance policies to all gateways, allowing you to manage API consumption, observability, protection, security, and more.

Important management policies you should apply through API gateways include:

- **Rate limiting and access quotas.** Setting usage limits for API access and controlling the request rate can prevent abuse.
- **Load balancing and circuit breakers.** Distributing incoming API requests; across multiple back-end instances; can support even load distribution, enhance performance, and help mitigate DDoS attacks.
- **Authentication and authorization.** Gateways can help secure access to managed APIs, while also enabling you to retrofit legacy back-end APIs with modern authentication and authorization mechanisms.
- **Content and schema validation.** Checking whether API requests and responses comply with contracts defined in the associated API specification can help guard against SQL injection, excessive requests, data exfiltration, request manipulation, or injection attacks.
- **Caching.** Within the security context, controlling caching behavior, including full and fragment caching options, allows you to reduce strain on back-end services, minimizing DDoS and related load-based vulnerabilities.
- **Monitoring and observability.** Centrally applied logging and monitoring mechanisms let you better understand how your APIs are working.
- **Version management.** Routing requests to the appropriate back-end API version can help you better manage deployments, while helping to minimize the risk of service disruption. It can also help you deprecate old API versions to reduce your potential attack surface.

API MANAGEMENT ON THE AZURE PLATFORM

The Azure API Management platform is a cloud-based service that enables API governance and discovery for not only internal APIs but also any external services that you consume, such as payment processing, social media, and cloud storage APIs. Its management capabilities complement the API governance and compliance capabilities of Azure API Center, and it integrates with the rest of the monitoring, security, and management infrastructure offered by the wider Azure cloud platform.

Chapter 5:

Fortify your APIs with advanced security solutions

Fortify your APIs with advanced security solutions

An API management platform helps you to gain visibility into your dynamic API inventory and to centrally apply controls and policies to your endpoints. However, designing a comprehensive API security strategy requires more.

Modern APIs face unique threats, and traditional rule-based security tools, like web application firewalls (WAFs) may not, by themselves, be fully capable of dealing with these threats. Many API specific risks involve business logic abuse, unintentional data exposure, or most commonly, broken or misconfigured authentication and authorization mechanisms.

The following entries from the OWASP API Security Top 10 list describe some of the most common examples of these types of risks:

- API1:2023 Broken Object Level Authorization. This is among the most common API vulnerabilities, occurring when APIs fail to properly enforce permissions for object access. For example, it allows a signed in attacker to manipulate request parameters, like user or object IDs, to access other users' sensitive data.
- API2:2023 Broken Authentication. This vulnerability is related to any weakness in the API authentication mechanisms, allowing attackers to exploit compromised authentication tokens, credentials, passwords, or other flaws to gain unauthorized access to your APIs.
- API3:2023 Broken Object Property Level Authorization. This risk involves incorrect access permissions to the APIs' available data or functionality. APIs with this vulnerability can return data to users who should not have access to it or can incorrectly allow users to add or delete data.
- API6:2023 Unrestricted Access to Sensitive Business Flows. These vulnerabilities allow normally legitimate API calls to be abused—for example, allowing an attacker to misuse a reservation system to reserve all available slots, preventing other customers from using the system.

To protect against these types of evolving threats, you need to go beyond static code reviews and rule-based security controls. You need the ability to fully understand how your applications and APIs are supposed to work, to detect when anomalous or malicious activity occurs, and to react quickly as soon as attacks are detected.

5.1 Integrate a CNAPP into your API security strategy

A cloud-native application protection platform (CNAPP) integrates multiple complementary cloud security solutions while providing users with a single integrated management interface. It can give you a holistic view of the security status of your applications, including information on your APIs, compute resources, back-end data stores, and all the other resources your application depends on.

While standalone API-focused security solutions can highlight specific vulnerabilities that your APIs face in isolation, CNAPPs offer much more comprehensive visibility into threats to your APIs. A CNAPP allows you to better understand your API risk surface in the wider security context of the applications, services, and infrastructure that your APIs interact with. This comprehensive view makes it easier for you to protect your APIs, prevent or detect vulnerabilities, and respond to emerging cloud security threats.

In the specific context of API security, a CNAPP can often provide security recommendations, vulnerability protection, threat detection, and response coverage for your APIs and any other resources that make up your cloud-native application. When paired with your API management platform, a CNAPP can help you understand the risk profile of managed APIs and harden them against attacks.

A CNAPP usually consists of several core capabilities that help boost your API security posture and protect your cloud workloads by:

- **Improving your security posture** using integrated monitoring, permission management, compliance monitoring, and risk identification capabilities to help identify vulnerabilities in your APIs.
- **Securing your data** by identifying the data stores that your APIs interact with and allowing you to classify sensitive data according to your organization's security policies and to apply compliance monitoring.
- **Securing your development processes** by embedded testing and security checks into your DevOps pipelines, helping to ensure that the API is validated as compliant with your policies and has been tested against known vulnerabilities before it can be deployed.
- **Actively protecting your APIs** by using runtime threat detection and protection capabilities, continuously scanning your APIs and their underlying infrastructure for potential vulnerabilities.

5.2 Monitor and protect APIs against attacks in runtime

Understanding expected API behavior and then monitoring real-time usage data is often the best way to detect active attacks. For example, after establishing a baseline number of requests per hour, seeing a 1000% increase in that rate could indicate a denial-of-service attack. In a more subtle scenario, a sudden increase in API calls where a single user is submitting multiple different user IDs may indicate an attempt to access protected data by using broken object level authorization.

While brute force attacks may be easy to detect using traditional monitoring systems, threats that manipulate an API's underlying request or authentication mechanisms are much more difficult to spot. Up-to-date global threat intelligence data, coupled with integrated AI and machine learning capabilities to learn your expected API usage, can help detect anomalous behavior and identify indicators of compromise (IOCs) as they happen.

A CNAPP can use these intelligent runtime monitoring and detection capabilities to triage the type and severity of detected issues, add them to tracking systems, and alert security teams. CNAPPs can also be used to trigger automated mitigation systems that can deal with a detected threat before it can do significant damage.

Combining the security and threat detection capabilities of a CNAPP with the management, controls, and governance capabilities of an API management platform allows you to build a comprehensive approach to API security.

CNAPP ON THE AZURE PLATFORM

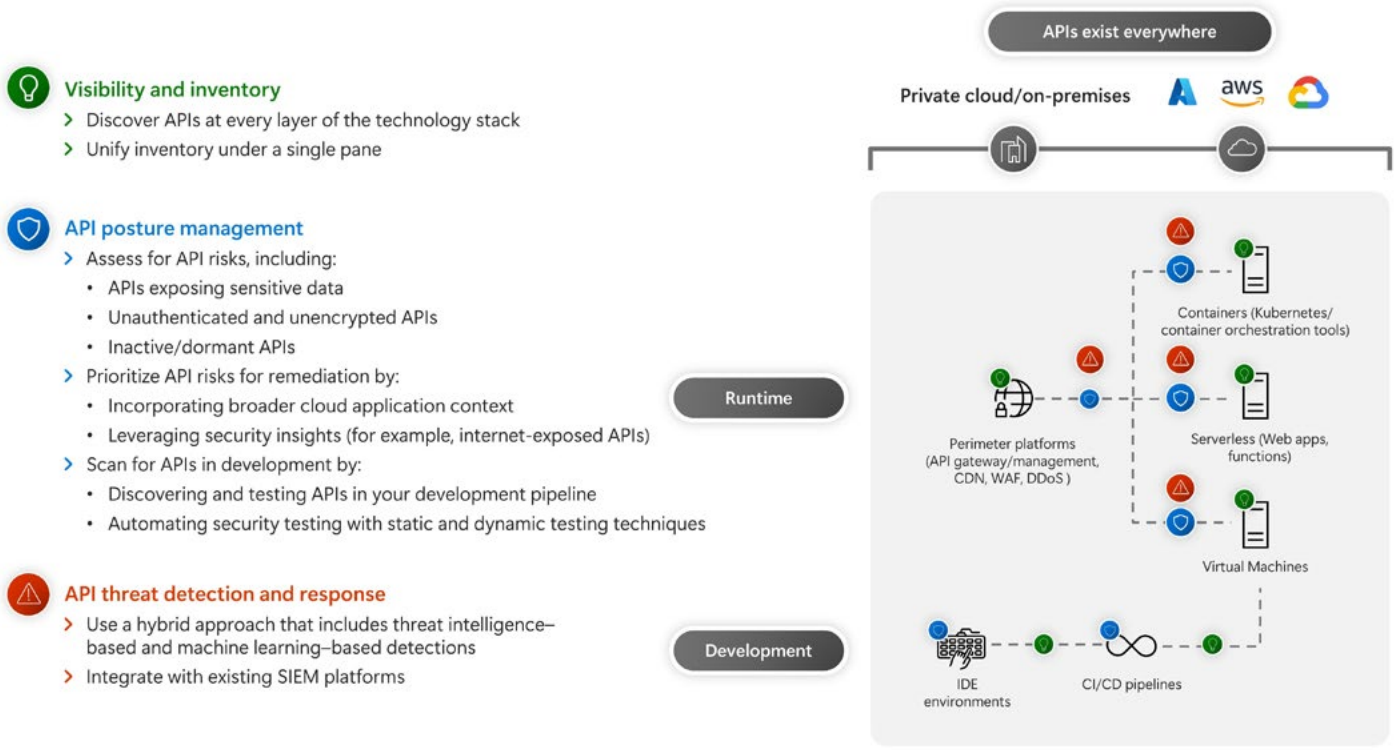
Microsoft Defender for Cloud is a comprehensive CNAPP solution, helping you enable secure development, continuously reduce risk, and remediate threats faster. It can help you gain visibility into your business-critical APIs, investigate and improve your API security posture, prioritize vulnerability fixes, and quickly detect active real-time threats.

Chapter 6:

Build layered defenses to maximize your API security

Build layered defenses to maximize your API security

An API management platform, coupled with the comprehensive protection of a CNAPP solution, can significantly improve your organization's API security. A layered API security strategy, built up using complementary security tools and services, gives you a stronger security posture, with additional ways to detect risks, prioritize threats, and reduce reduce the likelihood of an attack.



6.1 Build secure networks

Network security is a critical part of any API security strategy. Some of the major features to consider integrating into your combined security solution include:

- **Network segmentation and isolation.** Firewalls, virtual private networks (VPNs), and perimeter networks can further isolate access to specific subnets and ensure only authorized users or services can access your APIs.
- **Perimeter network devices.** Intrusion detection/intrusion prevention systems, such as application gateways and WAFs, can help protect APIs from common threats by filtering and monitoring HTTP traffic for malicious activity.
- **DDoS protection services.** Enterprise scale DDoS protection services help maintain service availability and business continuity, even during large DDoS attacks. In addition, they employ policies at the gateway layers with rate limiting and API request throttling to prevent excess traffic exposure.

Using the Azure cloud platform as an example of the types of tools and services that can help, Azure Virtual Network lets you impose network isolation requirements on your APIs by configuring the subnet and network routing they use. It also offers private connection methods, such as Azure ExpressRoute or Azure VPN Gateway, to securely connect your cloud-based APIs to on-premises or other external networks. In addition, Virtual Network supports Azure DDoS Protection for mitigating network and transport-layer DDoS attacks.

Azure Web Application Firewall further helps to protect your applications from web attacks, malicious bots, and application DDOS attacks. It also integrates with other Azure edge services, such as Azure Application Gateway and Azure Front Door. These capabilities complement Azure API Management and Microsoft Defender for Cloud to help build a solid defense-in-depth network security strategy.

6.2 Control and manage access

Identity and access management (IAM) systems help you control which users or systems can access an API, what actions they can perform on it, and how they are authenticated. When considering your API security strategy, make sure you look for how best to take advantage of each of these IAM capabilities:

- **Authentication.** Secure sign-in and authentication methods using multifactor authentication, single sign-on (SSO), and OAuth or JWT token-based authentication can help secure access to your APIs.

- **Authorization and role-based access control (RBAC).** Fine-grained access permissions based on user roles help ensure that users only have access to the APIs and data they need.
- **Access auditing.** Tracking and logging API access helps you detect and investigate security incidents, while also identifying wider data security and privacy compliance issues.

Again, taking the Azure ecosystem as an example, Microsoft Entra ID can provide centralized identity and access management services for all Azure-managed resources. This allows you to apply centralized role-based access controls and policy to control access to APIs. You can also use integrated Microsoft Entra ID–enabled OAuth authentications to secure access to your APIs.

6.3 Keep your secrets secret

Exposure of sensitive security information, such as API keys, passwords, tokens, and certificates, can leave your APIs and data completely exposed to cyberattackers. Consider the following advice to help ensure that your API secrets are as secure as possible:

- **Use secure storage.** Store your sensitive security data in dedicated storage vaults specifically designed to protect and manage secrets. Ensure that these vaults have fine-grained access controls in place, allowing only authorized users and services to access or manage your secrets.
- **Use automated rotation.** Implement automated renewal and rotation systems for your API credentials and certificates to minimize the risk of exposure from compromised or outdated information.
- **Implement auditing.** Maintain access and change tracking for your secrets to enable security audits, compliance verification, and incident investigations.
- **Store secrets in Azure Key Vault.** This service is an example of a secure storage location dedicated to secrets management. It has built-in features to securely store secrets, create and control encryption keys, and easily provision, manage, deploy, and rotate certificates.

A holistic approach to your API security strategy can minimize risks by taking advantage of multiple technologies and services. Combining networking security, access controls, and secret management with an API management platform and CNAPP allows you to build a layered, defense-in-depth API security strategy capable of keeping up with a constantly evolving threat landscape.

Chapter 7:

Next steps

Next steps

As you start to develop your API security strategy, review and explore each of these areas to be sure that you're applying best practices and employing the right tools to minimize API risks:

- **Discover your APIs.** Identify, document, and understand all of the APIs that your organization uses, and keep that information up to date.
- **Apply a centralized API management strategy.** Adopt an API management platform to help you secure and govern your APIs. This gives you a comprehensive set of tools and processes that help you design, deploy, monitor, secure, and analyze your organization's API usage.
- **Assess and improve your API security posture.** You need tools to help you perform ongoing testing, identify potential security vulnerabilities, recommend fixes, and quickly detect active real-time threats. A CNAPP can help you with tasks such as full lifecycle protection, vulnerability detection, and response coverage for APIs.
- **Monitor and protect your APIs.** A CNAPP can also help monitor your APIs, allowing you to understand and model expected functionality and behavior. Advanced machine learning capabilities help identify threats and detect attacks as they happen, and they integrate with your wider monitoring, alerting, and remediation systems.
- **Build a holistic API security strategy.** Take a layered, defense-in-depth approach to building an API security strategy. Use complementary tools and services to minimize your security vulnerabilities, and build a combined solution that best protects your APIs and applications.

7.1 Better together

Azure API Center, Azure API Management, and Microsoft Defender for Cloud work better together to strengthen your API security posture and minimize the risk surface that your APIs present.

- Azure API Center allows you to create and maintain a centralized inventory of your organization's APIs, across types, lifecycle stages, and deployment location.
- Azure API Management is a hybrid, multicloud management platform for APIs, across all environments. As a platform as a service (PaaS), API Management supports the complete API lifecycle.
- Microsoft Defender for Cloud is an integrated CNAPP that delivers comprehensive security and compliance, from code to runtime, for multicloud and hybrid environments. Organizations can develop and deploy applications securely, minimize risks with continuous posture management, and help protect workloads and applications from modern threats.

TRY MICROSOFT DEFENDER FOR CLOUD

You can try Defender for Cloud for free for 30 days with your Azure subscription.