neosec

The Do's and Don'ts of Modern API Security

Eight Critical Factors That Will Make or Break Your API Security Posture

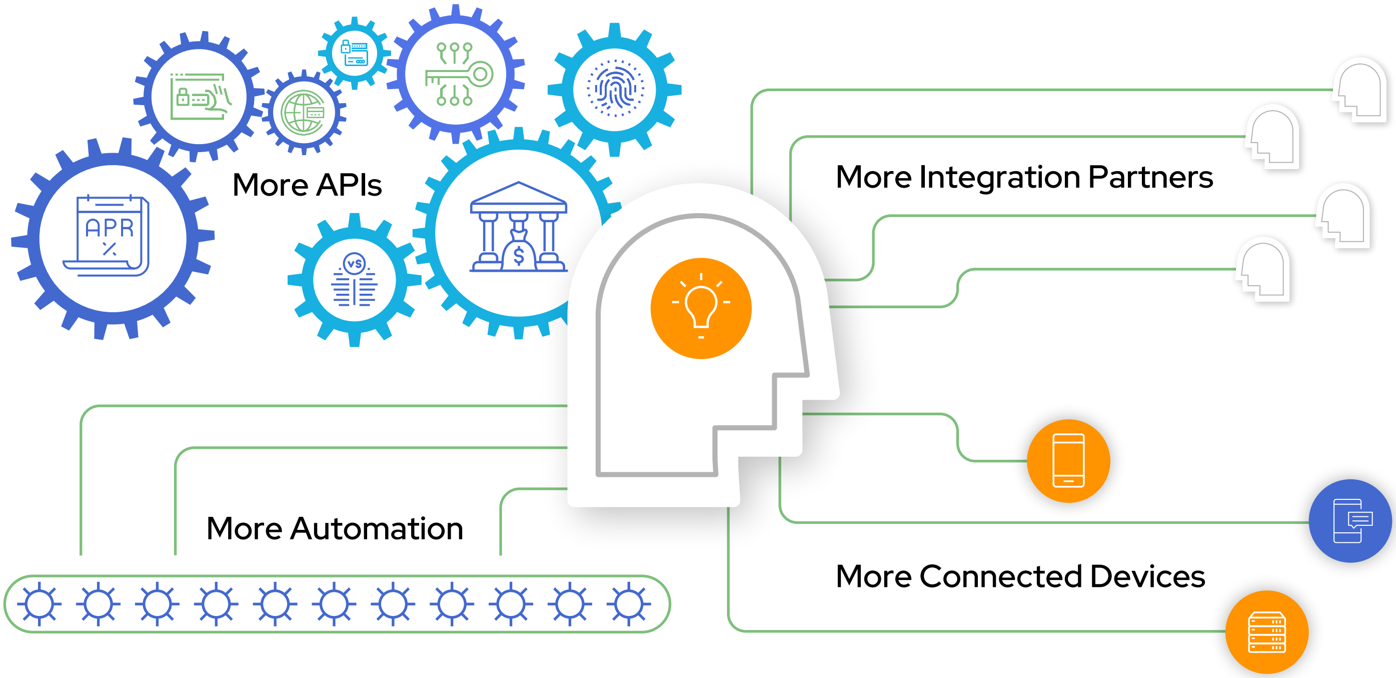
neosec.com

API security is a fast-growing priority

API security is shooting to the top of the priority list for many IT executives – and with good reason.

Consider that according to Gartner, "API abuses will move from an infrequent to the most-frequent attack vector, resulting in data breaches for enterprise web applications." In addition, look at what is driving the growth in API risk - more APIs, more automation, more integrations partners, and more connected devices.

What's driving this growth in API risk?



Now that users access critical application functionality and data in more ways than ever, the need for programmatic interfaces to critical application functionality and data is growing.

Application developers and API teams are innovating rapidly to address these new business needs, but too often security and governance of APIs lags behind.

This disconnect hasn't gone unnoticed by bad actors, who are moving quickly to exploit it.

> "API abuses will move from an infrequent to the most-frequent attack vector, resulting in data breaches for enterprise web applications."
>
> **Gartner**

So what’s so complicated about protecting APIs?

1 APIs are a moving target

Most security teams don’t have a complete inventory of the APIs their organization is exposing to the outside world. APIs are created and decommissioned on a near-continuous basis as a byproduct of fast-moving DevOps processes. Meanwhile, immature API practices lead to unintended exposure of sensitive APIs to external parties, including many “shadow APIs” that that are unknown to the security team..

> “API security is complicated by the fact that many organizations don’t have an inventory of the APIs they provide, or the APIs they use.”
>
> **“API Security: What You Need to Do to Protect Your APIs,” Gartner**

Without effective coordination between the teams creating and deploying APIs and the security teams tasked with protecting them, understanding and reducing the API attack surface is nearly impossible.

2 APIs are vulnerable to two very different types of threats

Another unique challenge of API security is that it must include measures that detect and mitigate attempts to exploit both technical vulnerabilities and loopholes in business logic.

Technical Vulnerabilities

Software vulnerabilities and misconfigurations, including the OWASP API Security Top 10, that attackers can attempt to exploit.

Misuse and Abuse

Business logic abuse and other behavior, like aggressive data scraping, that doesn’t necessarily exploit a technical vulnerability but nonetheless represents a significant business risk.

As security teams race to understand the full scope of their API attack surface and develop risk mitigation strategies, they must consider and plan for these two very different attack vectors.

The Do’s and Don’ts of Effective API Security

Addressing the complex challenge of API security requires a well-considered approach that:

Incorporates the latest technology advances

Breaks down organizational barriers

Addresses the complete API threat landscape

The following are some essential strategies – and pitfalls to avoid – as you develop a more sophisticated API security strategy for your organization.

Do’s	Don’ts
✓ Do approach API security as a continuous lifecycle	✗ Don’t be afraid of the cloud
✓ Do strive for the broadest visibility possible	✗ Don’t fall into the technology lock-in trap
✓ Do make business context central to your strategy	✗ Don’t make automation a one-way street
✓ Do make cross-departmental collaboration a priority	✗ Don’t accept a “black box” approach to API security

We’ll take a closer look at each of these topics in the pages that follow.

✓ **Do** Approach API Security as a Continuous Lifecycle

Many security teams begin their API protection efforts by creating an inventory of APIs in use by their organization. This is the right place to start, but one critical error that many teams make is approaching it as a point-in-time exercise.

Because APIs are added and decommissioned continuously, it's critical for security teams to keep a living inventory of API interfaces into their sensitive applications and data repositories. This should include the ability to discover the presence of new APIs, even if they weren't proactively communicated to the security team by the development or operations teams.

Continuous API discovery eliminates blind spots and ensures that the security team always has a complete picture of the attack surface that they are charged with defending.

When continuous API discovery is performed effectively, shadow, rogue, forgotten, zombie, orphaned, and deprecated APIs all become problems of the past, and security teams gain the visibility they need to detect and mitigate a wide range of emerging API security threats.

✗ **Don't** Be Afraid of the Cloud

As with many areas of information security, early attempts at API security focused on detecting known attack patterns through the use of signatures. This approach is limited and misses behaviors that don't have a known or easily detectable signature.

Attackers are increasingly skilled at circumventing traditional security measures, either by performing "low and slow" attacks that remain under the radar or using creative techniques like business logic abuse to achieve their objectives. The only way to defend APIs from the full array of possible risks is by tapping into the power of behavioral analytics using the latest machine learning (ML) techniques.

Security teams are sometimes reluctant to send sensitive information about their organization's activity to the cloud. However, performing true behavioral analytics using data science and ML on the volume of API data that most enterprises generate is highly impractical without the scale and elasticity that the cloud provides.

In addition, as security teams see their limited resources stretched thin, long and complex product deployments are a major obstacle to progress. Given the growing risk posed by broader API usage, security teams can't afford to fall further behind. As attackers increasingly tap into the power of automation, API defenders must automate and scale their techniques as well.

Therefore, it's essential to take the leap to the cloud as part of your API security strategy.

In addition to accelerating time to impact, cloud-based API security approaches make behavioral analytics at scale realistic and attainable for any size organization.

This doesn't mean that you should have blind faith that your data will be protected. Challenge your API security partners to demonstrate that your sensitive data will be both anonymized and protected in the cloud.

✓ **Do** Strive for the Broadest Visibility Possible

Another key shortcoming of first-generation API security techniques is that the scope of their monitoring and analysis is too limited across two very important dimensions:

Limited Time Horizon

Many API security products focus on monitoring individual API calls or, at best, short-term session activity. This isn't sufficient.

Many legitimate business processes occur over a much longer time horizon. Many attacks do as well.

Blind Spots

Approaches that rely exclusively on deployment of per-app sensors inevitably have gaps in API threat visibility.

Forgotten legacy APIs will never be discovered. As shadow APIs appear, it can't be assumed that a host-based agent will be present to detect them.

Overcoming both of these limitations is essential to stay ahead of today's more sophisticated API security threats.

In order to be effective, behavioral analytics for API usage must be performed over a period of at least 30 days. This provides a more complete and accurate picture of baseline expected behavior. It also makes it possible to detect attacks that are executed slowly across multiple days or weeks – and numerous API sessions.

Eliminating blind spots is also critical. The best way to do this is by ensuring that your API security platform can ingest information from the broadest possible range of data sources, including API gateways, network devices, microservices orchestration solutions, cloud providers, and more.

✗ **Don't** Fall Into the Technology Lock-In Trap

Another common pitfall to avoid with your API security strategy is linking your approach too closely with any one piece of your architecture. As desirable as it is to standardize, most organizations find themselves with a blend of different API technologies and supporting infrastructure – both on-premises and in the cloud.

For example, just because a centralized API gateway strategy is implemented, it can't be assumed that shadow APIs won't appear that circumvent the core API governance approach. It's also possible that different departments or business units will have legitimate reasons for using different API approaches and technologies.

The same is true when it comes to underlying infrastructure technologies. For example, if your API security approach is too entangled with the Amazon Web Services (AWS) cloud infrastructure, how will you adapt if Microsoft Azure or another platform is introduced as a result of an merger or acquisition or for cost and redundancy reasons?

Your API protection strategy must tie in with your primary API technologies, like API gateways, while also collecting as much information as possible from other sources such as network devices, cloud platforms, and microservices orchestration tools. This is the only way to create a complete picture of your API attack surface and future-proof your security strategy as technology and infrastructure transitions inevitably occur.

✓ **Do** Make Business Context Central to Your Strategy

Discovering APIs and identifying security risks are just the beginning of the journey to a smaller API attack surface. The real value comes with understanding the business purpose and functionality of each API. This isn't easy to accomplish, but it's a critical element of a modern API security approach.

Consider the following scenarios:

1. How do you know if the API credentials of a specific partner have been compromised?
2. How do you know if corporate espionage is happening in the form of data scraping on an API?
3. How would you know if your invoicing API is being abused by a user enumerating through invoice numbers to steal account data?

In the first scenario, activity would appear to be originating from a legitimate user. Therefore, the only way to detect malicious intent is by detecting a change from expected behavior on the API in question.

The second and third scenarios are also examples of unsanctioned behavior that exploits legitimate API access models. These are more cases where understanding of business context in addition to what is occurring technically is critical.

To be effective, API security techniques must move beyond technical elements like IP addresses and API tokens. Visibility into more business-relevant entities tells a bigger story. For example, knowing which MerchantID, AccountID, or other business process entity (e.g., invoicing, payment, ordering, etc.) is abusing the API helps identify issues faster and with fewer alerts.

✗ **Don't** Make Automation a One-Way Street

One of the fundamental capabilities of an effective API security approach is the ability to send alerts and events to preferred security monitoring and security orchestration, automation, and response (SOAR) tools. A common mistake made by security vendors – and the teams that implement them – is to view security alerts and automated responses as a one-way communication flow.

An alert without supporting details arguably does more harm than good. An alert with rich context about the cause and impact is much more actionable. But the real win comes from providing a context-rich, actionable alert and giving the receiver the ability to query a more extensive data set to analyze the incident.

Providing this additional capability helps security teams avoid alert fatigue and drive an effective incident response program. This is true for many types of security monitoring, but it's particularly important for API security, where the lines between legitimate and unsanctioned activity can often blur.

Similarly, initiating SOAR playbooks and workflows based on detected API threat activity is useful, but the value is increased by orders of magnitude when these tools can pull data specific to your organization's unique needs through a security API.

Do Make Cross-Departmental Collaboration a Priority

To be effective, an API security strategy must extend beyond inventory and reactive security monitoring. As important as monitoring and response capabilities are, the biggest API security gains can come from proactive avoidance of vulnerabilities during the design, development, and deployment phases.
Giving API teams visibility into how APIs are being used (and abused) under real-world conditions will help achieve buy-in for your security strategy. Over time, this exposure will foster a culture of thinking about security earlier during the API development and deployment processes.

That said, these efforts will only be successful if you:

- ✓ Make sure there are non-security benefits that help API teams work more effectively in addition to the core security features of your approach
- ✓ Make API inventory and activity information easy for non-security users to view and query
- ✓ Build in contextual responses such as integrations into development tools like Jira that proactively open tickets for security fixes that developers need to make

Thinking about API security as everyone's job and making it easy for stakeholders outside the security team to engage eliminates finger-pointing and makes it possible for development, operations, and security teams to work together in mutually beneficial ways.

> Giving API teams visibility into how APIs are being used (and abused) under real-world conditions will help achieve buy-in for your security strategy.

☒ **Don't** Accept a 'Black Box' Approach to API Security

Many security products lack transparency into the underlying data they use to detect possible threats. They treat this aspect of what they do as a "black box" that the teams using the product aren't granted direct visibility into.

The problem with this is that it limits the value of the information that is provided to the security team.

Requiring full data transparency from your API security vendor will give you two key advantages:

1 More actionable alerts

Even if an alert is accurate and includes useful supporting details, that is never the end of the story. There is always the need for further investigation and the development of response and containment strategies. If your API security partner gives you the ability to query their source data and ML model outputs, you'll be able to gain deeper insights into the "why" behind each alert. This is invaluable during the investigation and response process.

Greater transparency into what is wrong – and how to fix it – also allows security teams to empower engineers to address root issues faster and more efficiently. This can help your organization spend less time in reactive mode by making proactive changes that will avoid future security incidents.

2 Proactive threat hunting capabilities

While responding quickly and effectively to alerts is great, avoiding them altogether is even better. This is accomplished through proactive threat hunting. If your API security partner empowers you to perform data queries, you'll be able to test your own hypotheses, understand relationships, and identify potential threats before they escalate into a security incident. For example, if you identify a bad API usage behavior by a specific partner, you can look for similar behavior by other partners or suppliers with a few clicks.

Ideally, these type of rich query capabilities should be made available to you in two ways:

- A simple and intuitive user web interface
- A set of API interfaces into the API security provider themselves for use in developing more sophisticated workflows.

Get Started Today

Neosec's cloud-based approach makes it easy to get started in minutes.
Within hours, you'll have a complete picture of API usage across your organization, including a detailed understanding of the relationships between your business logic and your APIs.

Are you ready to take the first step to a modern, systematic approach to API security? **Visit Neosec.com to start your free trial.**

neosec