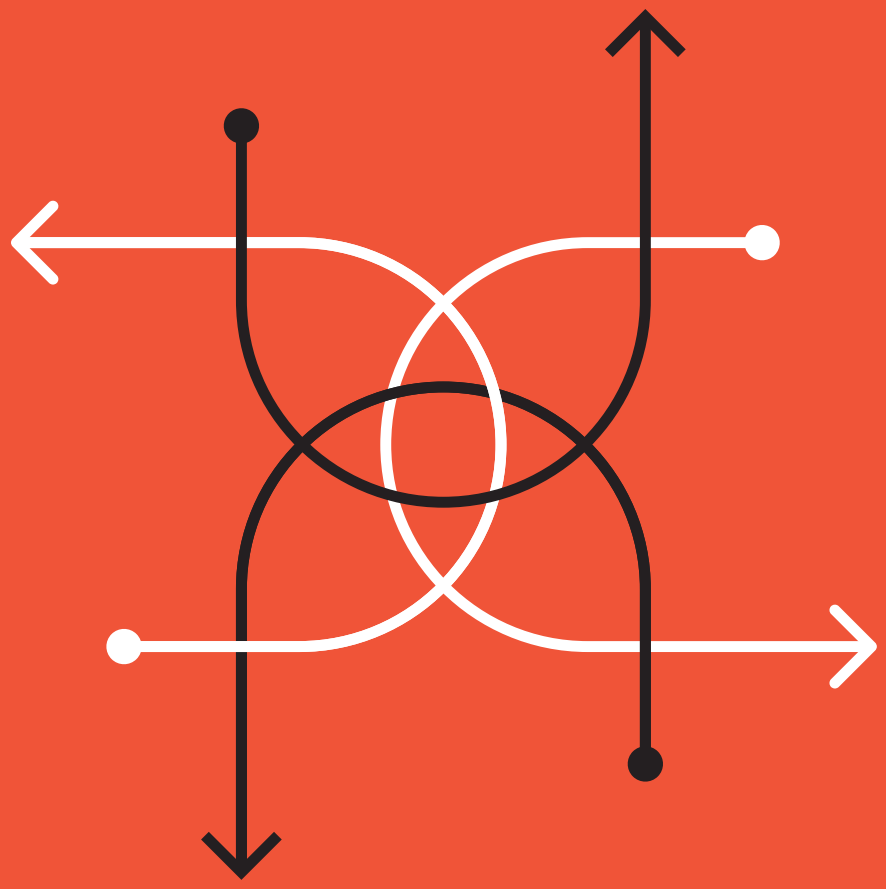
The Practical Guide to API Security

By Aaron Lieberman



BIG COMPASS

> "APIs are rapidly becoming one of the most important infrastructural layers of the Internet. They are difficult to secure and determined hackers are extremely tenacious in finding ways to exploit them."
>
> *– David Berlind, ProgrammableWeb Editor in Chief*

Executive Summary

APIs are fast becoming the de facto tool for connecting businesses, eliminating silos within the enterprise, safely exposing data, and breaking down monolithic applications into maintainable and reusable services.

As the use of APIs has increased, however, so has the attack surface of your system and therefore the risk that they pose. In the rush to realize the benefits of these services, many organizations sideline or forget certain crucial processes.

One of the most critical disciplines that fall to the side is API security. Obviously, API security is an area that can't be ignored. Gartner predicts that APIs will become the most significant security concern for enterprises in the near future. Several large organizations – including Facebook, T-Mobile, and the USPS – have been victims to breaches that originated with vulnerable APIs.

One of the strengths of APIs is that they utilize the technologies that the Web was built upon. Unfortunately, the same strategies that are used for securing web applications will not suffice for APIs. A different approach is needed.

That approach spans everything from establishing ownership and roles to implementing the right tools and best practices. The skills, responsibilities, and processes for creating secure APIs requires a multi-disciplinary approach and a balance between what is needed to protect the organization's systems and data and how to exploit the benefits of APIs to the business's advantage.

In the end, API security should be top of mind, and the best way to start to secure your APIs is to put some focus on securing them. it may not be as difficult as you think, as there are straightforward steps you can take to begin to secure your APIs. In this guide, we've compiled the knowledge that Big Compass has gained while working with organizations, large and small, and at every point along the API adoption spectrum.

Audience

This eBook is intended for developers, technical leads, architects, and IT and tech-savvy business leaders interested in building and securing a robust API environment. At the end of this eBook, you'll have a practical understanding of how to assess your API security posture, what needs to be done to implement API security, and a vision of how to mature your security as your enterprise's use of APIs increases.

While this eBook is not a hands-on guide to implementing API security for your business, it does require a basic understanding of APIs. It's also helpful if the reader has at least a passing familiarity with APIs, microservices, integration platforms, and the advantages of DevOps.

Why you should read this ebook

"APIs are one of the most valuable assets an organization can develop for numerous reasons. With the rise of APIs, the trend of advanced attacks and insider threats is only increasing though. So, it is more important than ever to make sure that your organization has complete visibility into all active APIs, that it can track all API traffic (on a per user basis) and can flag any abnormal situation."

– Bernard Harguindeguy; CTO, Ping Identity

"One of the most fascinating things about APIs is their generality. A single API can be used by different roles (guests, regular users and admins) and different clients. It saves backend developers time and effort, but also makes it much harder to implement an effective access control mechanism. Almost every API breach we see on the news is related to access control issues. Companies should invest more resources in planning and securing their most critical gates – authentication and authorization."

– Inon Shkedy; Security Researcher at Traceable, and API Security Project Leader at OWASP

"API security is often misunderstood in our industry. This eBook clearly outlines how to protect APIs by using a layered approach to API security. Teams tasked with developing APIs should read this eBook first to get a primer on what a good security posture should look like."

– Francois Lascelles; Field CTO, Ping Identity

Chapter 1

The Rise of the APIs... and Security Risks

APIs are the connectivity and functionality mechanisms with which enterprises can enable digital transformation. The stark growth in the number of APIs indicates how much enterprises and developers value the technology.

According to ProgrammableWeb, which chronicles the public API sector, the number of APIs has been increasing sharply since the late 2000s. API growth continues at an even faster pace and shows no signs of slowing.

As enterprises add more and more APIs into the mix, breaking down monolithic applications, accelerating communications with partners, and offering clients new and innovative services, they become increasingly dependent on many applications. Their size makes them no less important to the business than their larger, complex ancestors.

However, as APIs' deployment continues to increase, proper API security is not as widely practiced as it should be. The lack of API security awareness is concerning, as the rise in APIs means there is a corresponding rise in security risks for enterprises.

Consider some recent and high-profile API-based incidences at Facebook, the United States Postal Service, and Equifax. APIs, acting as the data-rich links between different applications, expose multiple vulnerabilities that can be targeted by hackers and extend the attack surface of an enterprise.

Approaching API Security as a Priority

What do we mean by API security? As detailed here, "API security describes the practices and products that prevent malicious attacks on, or misuse of, application program interfaces (API). API Security is part of API management and governance."

APIs use web technology to integrate applications, but it is a mistake to assume that APIs can be protected using the same practices and techniques used to secure the Web. The risk profile of APIs is entirely different and requires a different security approach. Developers who fail to use or write APIs with security as a central focus compromise both the data and the applications.

Enterprises have to assume a proactive approach in API security and prepare for the worst-case scenarios. The security of an API is one of the first things that should be established when developing that API's architecture. Security testing should begin early well before deployment and continue throughout subsequent development.

It is also during API development that decisions should be made regarding how specific requests should be handled. The security measures may be broad at first but should eventually be narrowed down based on the enterprise's needs.

"A growing API footprint is beneficial for digital transformation purposes, but it also means that your attack surface is increasing. It is imperative to make sure that your organization's valuable data is protected by implementing advanced API threat protection like PingIntelligence for APIs to detect and block advanced attacks that would previously go undetected for months or even years."

– Bernard Harguindeguy

How is API Security Being Handled?

There are many ways hackers are using APIs to gain access to enterprise systems. Some of the most common attack paths include parameter attacks (often via an SQL injection), identity attacks, and man-in-the-middle attacks. To combat these attacks, the most widely used API security models are employing identification, authentication, and authorization measures implemented with the use of tokens, API gateways, quotas, and throttling and encryption and signatures.

Some enterprises are opting to use in-house solutions for API security and are finding their efforts becoming mere reactionary measures to attacks rather than actively preventative ones. There may be no consensus between an enterprise's IT security team and the API development team on who is responsible for API security, which can cause the task to be delegated downward.

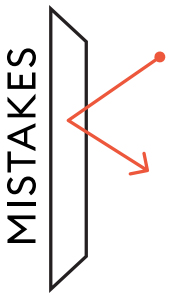
More enterprises are using API-security firms that have routinely updated threat databases and that offer a complete arsenal of identity and management tools. However, even with these tools, the protection level can fail to detect the most sophisticated attacks. While the security measures are robust, additional steps are needed to address the resulting security gaps that arise when APIs are deployed. This has presented an opening for applying machine learning-backed API security, an area to be further investigated and discussed.

Conclusion

API security has not kept pace with the broad adoption of the technology. It's crucial that businesses begin to take API security seriously. As we'll see in later chapters, different API use levels call for varying levels of scrutiny and intervention. Regardless of the level of security measures required or how they are employed, technology leaders and teams must understand the risks APIs can pose, along with the benefits, and be prepared to lay down a foundation for secure API development and use.

Chapter 2

Common API Security Mistakes

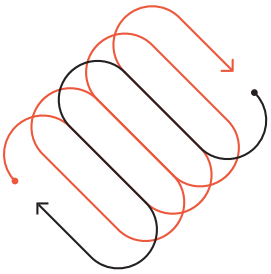


In the rush to adopt this new technology that promised – and delivered – greater flexibility and agility to businesses, companies began building APIs. Only later did some discover that, without process and planning, API implementations can get quickly out of hand. Others are still learning this. Unfortunately, security was one of the design process elements that can get sidelined in a company's early days of realizing the power APIs provide.

There is a distinct difference between merely building an API and designing and developing an API with the security features that allow it to address the needs of the users reliably. Here are some common API security mistakes to be aware of and helpful best practices for avoiding them.

> The first thing I do during a pentest, is to look for the most niche APIs that a company exposes. These APIs, that are barely accessed on a daily basis, are the most likely to be built without security in mind. Don't neglect the APIs that support non-prod environments, temporary features and older versions - there is a good chance that the next breach would come from them."
>
> *– Inon Shkedy, Security Researcher at Traceable, and API Security Project Leader at OWASP*

No Plan for API Security

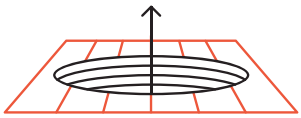


Not prioritizing API security during development can be attributed to apprehension about tackling API security, a lack of expertise on the subject, or reluctance to be held responsible for any security complications. Also, many times, organizations want to develop quickly, and API security is not seen as a priority to move a product to production. However, not having an action plan in place for worst-case scenarios could leave an enterprise severely compromised in the event of an attack.

MITIGATION:

Security should be a primary focus during the development of an API, not an afterthought. There are many different development tools available that can be used to ensure an API has the proper security features it needs.

Leaving APIs Exposed

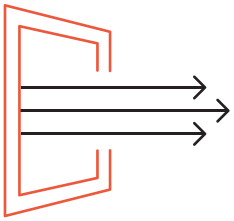


Implementing an API with no mechanisms in place for verifying who is trying to gain access and whether they have the appropriate authority exposes both the API and any connected digital assets to hackers and bots searching for vulnerabilities and will gain access to exposed APIs and mine for sensitive data.

MITIGATION:

Implement API security - even if it is minimal. This can include incorporating whitelisting so that only specific IP addresses are allowed. Basic authentication can be used at the front-end, requiring usernames and passwords. You can take security a step further by instituting multi-factor authentication at the front end followed by some form of authorization check so that the appropriate users have access to certain types of information. There also should be a checkpoint at the back end of an API.

Homegrown API Security

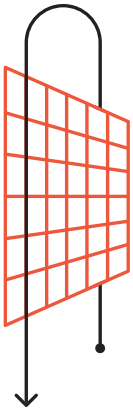


The risk with keeping API security in-house is that errors can occur, even if security was a priority during development. There can be loopholes in the code-API code that is poorly written could be easy to hack and is a security risk. Another factor to consider is the effort and time required by an enterprise's developers to ensure that security standards are updated frequently and are effective. This is likely to occur at the expense of other projects and result in technical debt.

MITIGATION:

Use third-party API security providers, like Ping Identity or Okta, that keep up to date with the latest in API security for you, so you don't have to. The security standards used by their platforms are routinely updated and enhanced to handle new threats.

Not Having an Organizational Governance Policy



API governance can be considered a form of API maintenance and is an essential part of API management. The governance policy should encompass documentation; if there is no documentation when APIs are initially set up, it can be challenging to go back to properly inventory them, particularly if an enterprise has many APIs. A governance policy is also necessary to track which APIs can be accessed by which parties, who owns the APIs, and who is responsible for maintaining them. Without a clear picture of its API environment, an enterprise can suffer from a lack of agility as it tries to function without knowing precisely what APIs they have, which ones are secured (if any) and who should have access to them.

MITIGATION:

Establish during the design and development processes who owns the APIs and are responsible for their security. Documenting is a key piece of proactive protection and reactive mitigation of security

risks, especially as an API evolves. Documenting the vast network of APIs in a complex organization may be a daunting task. Consider using a third-party API security and risk management provider, like Traceable Defense AI, that provides continuously updating automatic API documentation and tracking of API definition and usage.

Best Practice: Require a minimal level of API security. The minimum level of API security would include these three (3) basic security mechanisms. First, basic authentication can be used at the frontend requiring usernames and passwords. This mechanism has, over-time, become less and less recognized as a responsible and vigorous response. Second, incorporating Internet Protocol (IP) address whitelisting so only specific IP addresses may access the API. Third, authorization processes can segregate the originating request's data and information needs and allow appropriate access based on level access definitions.

Conclusion

API security shouldn't be an afterthought. Avoiding breaches requires that API security be an integral part of planning and design. The few mitigation actions in this chapter are easy to implement, but as we'll see in chapter 3, best practices for API security should be an ingrained element of the end-to-end process.

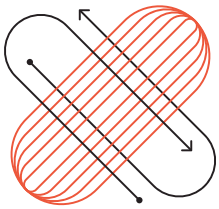
Chapter 3

Best Practices and Benefits of API Security

As discussed in the last chapter, the rush to realize the benefits of API-led integration frequently leads to API security being overlooked or shoved to the side. While organizations have the best intentions to revisit their API applications' security, setting this vital aspect of development aside for later can lead to severe consequences, especially for public facing APIs.

API security doesn't need to be overly complicated, but it can seem overwhelming to many, given the advanced threats that occur in the industry. Following a few best practices, API security will change from a daunting task to one that most organizations can accomplish.

Plan for API Ownership



Many times, we see companies implementing APIs but skipping the critical step of assigning an owner. Because we're talking about security, an API owner goes beyond who does the application's maintenance.

An owner is responsible, among other things, for ensuring API security is considered and managed. Because an owner must take charge when there is a security incident that involves the API, it's in their best interest to make sure securing the application is part of its development.

When planning or designing your API, you need to ask:

- Who or what group owns the API(s)?
- Who is responsible for maintaining it or them, including updating documentation?
- Who reacts to an API security incident?

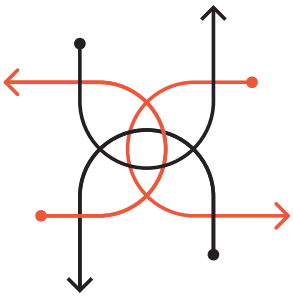
Ownership of an API is at a higher level than accountability or responsibility. An owner might even be a CTO. If it's a group, then it's typically the group leader or architect at the tip of the spear when there is an incident.

BENEFITS OF DEFINING AN OWNER

One of the benefits of identifying an API owner is it's clear who should be directing action during an incident and who is ultimately in charge of ensuring security is considered and implemented.

Without ownership, an incident can result in many different people and groups pointing fingers in many different directions. It's more likely that an API will be protected to the fullest from the start with ownership.

Include API Security in Your Planning or Design Phases (Don't Skip Out on API Security)



API security can fall victim to the desire to implement a solution quickly. At other times, API security ends up an afterthought. In both cases, you can get seriously burned by leaving your APIs unprotected.

API security requirements are business requirements. Since security is a business requirement, it is not optional. APIs are an essential part of your business results, so protecting your APIs is the same as a secure lock on a bank vault

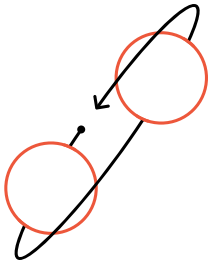
The surest way to prevent security from being sidelined is to bake it into the API's design and planning process. It can take some time to include security, but this is time well spent. Plus, including it in the design means you can choose the type of API security most appropriate to your application right from the start.

Depending on your environment and the application's purpose and availability, you may choose to implement only OAuth 2.0 and IP whitelisting. However, you may discover that your API needs greater protection and decide to include WAF, OAuth 2.0, IP whitelisting, and even Traceable or PingIntelligence for a public-facing, high-risk API. By including security thinking early, you'll have the ability to bake the right security model into the development process instead of retrofitting it later.

THE BENEFITS OF PLANNING AND DESIGNING API SECURITY FROM THE START

By raising the issue of API security from the beginning, it will raise stakeholder awareness of the need and keep it on their radar. Project milestones can be set with the development of the security components included. Stories and epics can be created alongside the other app features. Plus, the entire team will be bought into security as part of the API's creation.

Build-in API Monitoring



As much as possible, your security model should allow you to be proactive in protecting your API, not reactive. Building monitoring and alerting into the app promotes that aggressive stance.

During the design phase, you should ask:

- What kind of metrics do we want to see?
- Do we have the visibility needed for our use cases?
- Do we have the ability to detect a breach?

As you consider your metrics, start with the ones that will give you the alerts needed to address issues quickly. As a baseline, your metrics should include:

- Latency
- Request size
- Response size
- Geographic location

Alerting on these metrics will keep you aware of performance and outliers without requiring someone to look at a screen all day. These metrics can also help identify problems when the API performs poorly, which can be like trying to find a needle in a haystack.

But these metrics and their alerts may not be enough. They might inform you when specific attacks are underway but may not catch more sophisticated breach attempts.

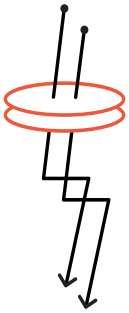
If you must detect more advanced attacks, you may need a more refined approach like machine learning or an AI engine. A product like PingIntelligence or Traceable can also help find API security-related attacks by learning your API's behavior. Using an AI/ML software to help protect your APIs also allows you to track metrics and usage patterns.

THE BENEFITS OF ALERTING AND MONITORING

Understanding typical performance and metric numbers will help you improve your API user experience. It also creates a viewport to detect breaches where you may not have had one before. Setting metric limits allow for alerting and can even kick off automated actions.

Metrics and alerting also free up time for your team. With well-defined metrics and the right tools, no one needs to spend time actively monitoring your APIs every day - the system can monitor itself and let you know when a problem or anomaly occurs. Tools like Traceable provide a great interface for monitoring and tracing the traffic on your APIs.

Protect Your API with SOMETHING
(Something is Better Than Nothing!)



This may not seem like a best practice, but when the alternative is to leave your APIs unprotected, something is better than nothing. That's because bots can and do scan the Web for open APIs, and they will find yours if it's left unsecured. Even the most basic security actions will prevent more automated and brute-force type attacks.

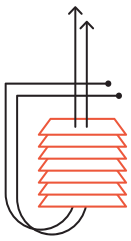
At a minimum, you should put API gateway security on your app, whether it's:

- IP whitelisting
- Basic authentication
- OAuth 2.0

Again, depending on your environment, whether your app is public or private, and your API's purpose, you may choose to go all the way, including OAuth 2.0, IP whitelisting, and operational tools such as PingIntelligence, Traceable Defense AI, and/or WAF. Do as much as is warranted by the API, its intended use, and its risk association. The benefits of doing somthing.

By doing even the minimum, you'll protect your APIs from common attacks, including bots probing the internet for low hanging fruit. With an even more secure API posture, you'll eliminate many sophisticated and known attacks.

Maximize Your API Security with the Layered Approach



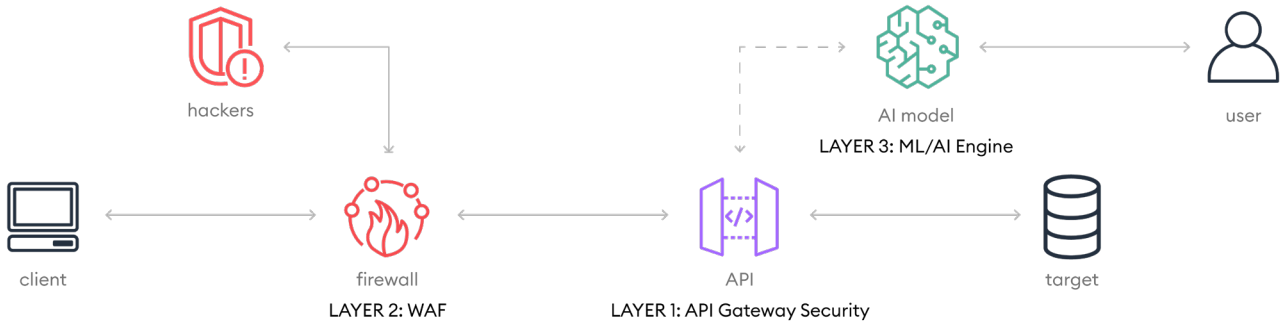
Do you want to sleep well at night? Then, using a layered API security approach is your best bet. As discussed over on Ping Identity's blog, this is the ideal way to protect your APIs.

Having secure APIs requires a multi-layered approach. Knowing how these three layers interact with one another to provide protection is essential for adequate API security.

The layers of API security include:

- API GATEWAY SECURITY:
  great for rate limiting and access control
- WEB APPLICATION FIREWALL (WAF):
  Great for OWASP top 10 protection
- MACHINE LANGUAGE/ARTIFICIAL INTELLIGENCE (ML/AI) ENGINE:
  Engines like PingIntelligence and Traceable monitor your APIs' behavior and protect against advanced, authenticated attacks that can fly under the radar

This simplified illustration gives you an idea of what layered API security looks like, conceptually:



THE BENEFITS OF LAYERED API SECURITY

Using a layered approach gives you the best protection with the most flexibility to address attacks. Using an ML/AI engine is like fighting fire with fire – hackers are using automated, AI-driven attacks, so your protection should, too. API security practitioners must use API gateway security to protect against standard attacks, WAFs to protect against OWASP top 10 attacks, and ML/AI engines to protect against advanced attacks using sophisticated methods.

Using ML/AI has additional benefits, as well. For instance, you'll gain deep visibility into your API's usage, operation, and performance. An ML/AI model will learn your API's normal operation and alert you if anything deviates from those norms. That data can also be mined for actionable insights on how to improve your application.

ML/AI models also offer a greater level of automated threat protection. There is no need to update security protocols or algorithms when using artificial intelligence and machine learning. A great in-depth read of layered API security can be found on Traceable's blog.

Conclusion

API security should be an integrated part of your API planning, design, and development process. If left as an afterthought, you're rolling the dice on not if, but when, your API will be found by bad actors and exploited. The benefits of securing your API are substantial and far outweigh the cost of developing with API security in mind. Following these best practices puts you in a far better position.

Of course, this means that you'll need to look closely at the composition of the teams responsible for API design, planning, and development to ensure that they have the tools they need to manage and implement security measures successfully.

Chapter 4

Must-Have Skills for Effective API Security

Gartner reports that by 2022, APIs will be the most frequent target hackers use to compromise data. API security, which should always be a concern of enterprises, requires a particular set of skills – skills that may not be fully expressed in your current teams. In fact, some of the skills that should be found within your API groups might be surprising.

However, these skills are necessary for developing and implementing the solutions and strategies that can adequately address the vulnerabilities and security risks unique to APIs.

Technical Skills

TECHNICAL AWARENESS

Having expertise in certain products or brands is not as important as having a thorough technical awareness. Technical awareness gives you a clear picture of the product and technology landscape and is more valuable for understanding and mitigating API security issues.

DESIGNING FOR API-LED CONNECTIVITY

With API-led connectivity, you can methodically connect data to applications by employing reusable and purposeful APIs. These APIs have to be designed and developed to properly execute the specific roles assigned to them and correctly fit into the security framework.

API DESIGN BEST PRACTICES

There are well-crafted design principles that can help make APIs more convenient for developers to consume. These best practices, which can include prioritizing documentation, improve the developer experience, and reduce incorrect code likelihood.

NETWORKING AND NETWORK SECURITY

Areas of vulnerabilities increase as networks, and network infrastructure capabilities continue to drive both application and API development. Knowing what is necessary for maintaining network security is critical.

API SECURITY BEST PRACTICES

Adhering to best practices for API security can help ensure that API deployments do not result in security issues. This can entail being mindful of the risks of APIs and carefully monitoring add-on software.

GENERAL AUTHENTICATION MECHANISMS

Being serious about API security means having a solid understanding of effective authentication methods, such as Oauth2.0, SAML, SSO, basic authentication, etc.

GENERAL AUTHENTICATION AND AUTHORIZATION FLOW

Securing authentication and authorization requires a careful understanding of API security strategies. Implementing access control using a modern API security strategy can be compared to the airport model, as explained here, where multiple techniques work in unison.

CYBER SECURITY THREAT AWARENESS

To stop hackers, you have to know how they implement their attacks. This entails having a working knowledge of:

- OWASP top 10 attacks
- Basic attacks
- Advanced attacks
- Attacks at different network layers
- OWASP API Top 10

→ OSAWP API Security Top 10

1. Broken Object Level Authorization
2. Broken User Authentication
3. Excessive Data Exposure
4. Lack of Resources & Rate Limiting
5. Broken Function Level Authorization
6. Mass Assignment
7. Security Misconfiguration
8. Injection
9. Improper Assets Management
10. Insufficient Logging & Monitoring

Non-Technical (Soft) Skills

EFFECTIVE COMMUNICATION

API development does not occur in a vacuum. Effective security requires communication with InfoSec, enterprise architects, developers, CIOs, and CISOs. Security professionals will have to bring all of these different people together and get the most out of each of their skills to help secure the API environment. You need to be able

to relay the purpose of the API and what data is exposed and articulate the data in a manner easy to understand for those who have to use and apply the information.

Effective communication is also necessary for the gathering of API requirements.

Post-development, API developers also have to communicate design and how to consume APIs. Visual communications, such as diagrams of different layers of security, system interactions, and data flow, can help people understand the API. Visual tools can also expose API definitions or communicate how to consume APIs, assisting consumers in testing the API's end-user version.

EMPATHY

Empathy helps to provide insight. Not understanding the consumer's perspective can contribute to conflict between the technical teams and users because the consumers want flexibility and greater access. You also need to understand APIs from the viewpoint of hackers to know how to stop them.

Empathy is gained through experience. Being curious and seeking knowledge by asking pertinent questions can also help you understand another point of view. How does the user interact with the API? How would a hacker interface with this API? What kind of information can a hacker get from the API? are examples of the type of thinking empathy allows.

FORESIGHT

You also have to keep in mind the future issues associated with APIs and what contingencies, plans, or solutions need to be in place for what will happen to the API in the future. Long-range issues can pertain to management, maintenance, the impact of personnel turnover, how to execute version control, and how to remain up to date with the latest in API security.

Conclusion

The skills needed for successful development and security of your APIs reaches across IT disciplines and roles. Of course, these different roles all approach a problem or project from different perspectives. That's precisely what you want, but at the same time requires a balancing act to achieve the business's goals while developing and maintaining secure APIs. In chapter 5, we look at this natural tension that must exist for API security.

Chapter 5

The Natural Tension that Exists in the World of API Security

Secure development of APIs comes from bringing together the right tools, teams, and skills, within a well-considered structure and process. Getting that all to come together well on any project requires a few things. One is that everyone understands and is committed to a common end goal. The other is a balance created between the collaborative but sometimes opposing forces required for success.

There are several areas of tension in the API security world. Exposing data and processes needs to be balanced with permission and credentials. The speed of development might be challenged with consistent implementation and standards.

For your enterprise's API security to be effective, it is important to plan who has to be involved and what responsibilities each party plays. That clarity will help focus on the natural tension of API security into productive areas that improve the end product while avoiding the more harmful elements that a lack of direction can create.

We're not the only ones that think this. Leaders in the API security space such as Traceable also agree.

Roles

The three principle entities that lead to top-notch development of API security are:

- Information Security Group
- API Developers
- Enterprise Architects

The operational side of monitoring and managing your APIs should also be mentioned right along with securing your APIs. The API operations team that governs your organization's APIs are a key part of helping you keep APIs up to date, managing APIs, and working with other teams if an attack or breach were to occur. Ensure that your API operations team is trained on managing your APIs and alerts notify this team so they can react to anomalies and escalate when necessary.

The CIO should define the relationships between the three entities in most cases. The primary benefit of having assigned and defined roles and having players understand their responsibilities is that cohesive relationships can be developed. These relationships will ensure the elements for a comprehensive API security environment are in place.

Typically, API developers will report to the CTO, the InfoSec group will report to the CISO, and enterprise architects are likely to report to either the CIO or CTO.

If any of the roles are absent or the players fail to understand or adequately execute their duties, there are likely gaps in your API security.

ROLES AND RESPONSIBILITIES OF THE INFORMATION SECURITY GROUP

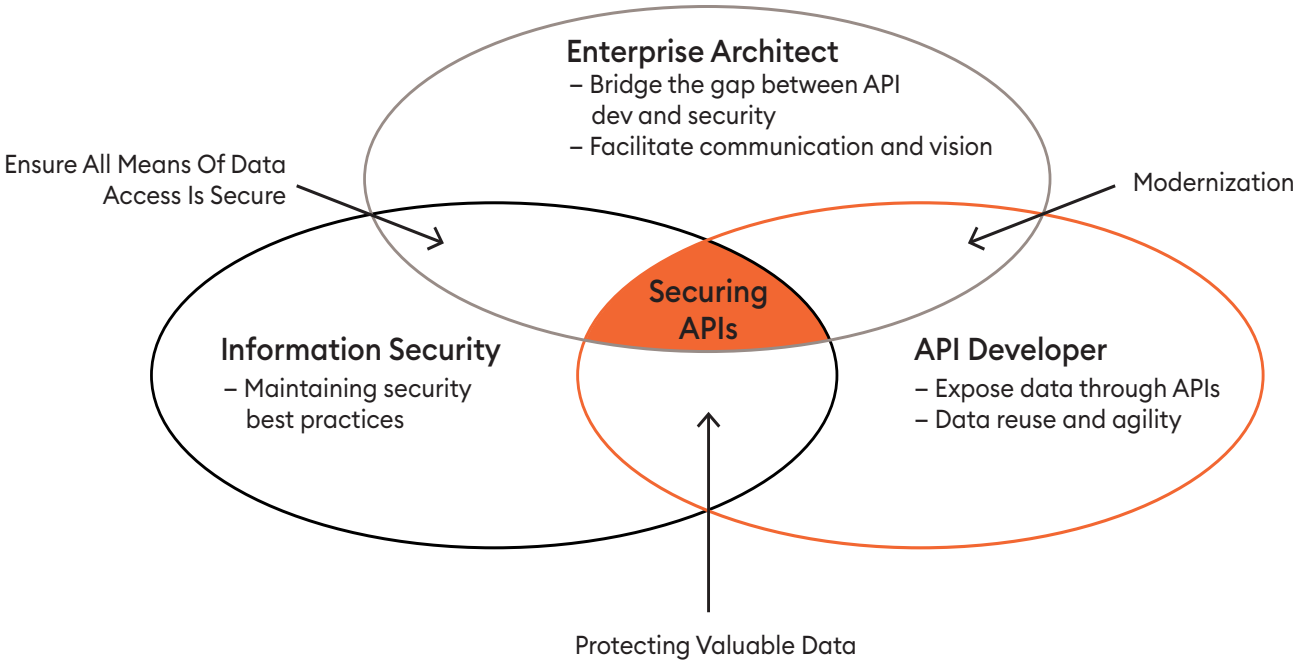
The information security team is tasked with securing and protecting the enterprise's data and systems. Its objectives, policies, and processes center on locking down the system and protecting valuable data.

ROLES AND RESPONSIBILITIES OF API DEVELOPERS IN API SECURITY

There tends to be a natural tension between API developers and information security teams because their goals and processes for handling data exist on opposite ends of the spectrum.

API developers are focused on exposing data through APIs for reuse and utilizing existing APIs. The developers leverage enterprise data via API to produce digital value.

There is also usually a significant role to play for API developers within API governance and security.



ROLES AND RESPONSIBILITIES
OF ENTERPRISE ARCHITECTS IN API SECURITY

Enterprise architects bridge the gap between the information security team and API Developers. They facilitate communication between both groups to gather sufficient information and insight to create API specifications and governance rules.

Architects have to determine how API security fits into the enterprise's overall security protocols and translate the requirements into the appropriate API architecture, ensuring that it is established, secure, but exposed. Additional responsibilities for Enterprise Architects include updating and modernizing APIs and developing API security rules as the enterprise matures.

The Risk of not Defining API Security Roles and Responsibilities

An enterprise without clearly defined roles and ownership over APIs will lead to gaps in its API security management. This is because API developers, who are focused on functionality and agility when it comes to APIs, will attempt to take on security tasks. Lacking the proper expertise, guesswork becomes a standard part of the enterprise's API security management, resulting in inconsistent management and insufficient security.

In the absence of enterprise architects, there can be a natural conflict between information security and API development teams just by defining the roles each entity plays in the grand scheme of API development.

The two teams have very different, contrasting objectives: information security protects data, while API developers expose data. Additionally, information security personnel tend to lack information that could be shared from an enterprise architect that would lead to adequately protecting APIs. Conversely, on the development side, security has not yet become a top priority for API development in

many enterprises. Without enterprise architecture, there is a lack of unified communication and effective collaboration between the two parties.

This can have a cascading negative impact, resulting in inadequately designed APIs, a lack of API ownership, and an increase in the rework and maintenance costs needed to fix gaps in APIs. Without the proper definition of these roles, this ultimately leads to developing APIs prone to attack by hackers.

Conclusion

For optimal API security within an enterprise, all three entities should be fulfilling their clearly defined roles and responsibilities. The absence of any one party, mainly the information security team or enterprise architects (API developers are always naturally necessary for developing APIs), results in the other party or parties assuming responsibilities they are ill-equipped to execute. If one or more roles are not present, organizations must take a serious look at filling roles or training a team to fulfill all of these vital roles in API development.

Chapter 6

API Security: Get Started Now

Over the last few chapters, we've made the importance of API security pretty clear, discussed the roles involved, and even outlined some of the best practices. Hopefully, at this point, you're fully on board with the need to secure your APIs. You probably have one remaining question, though.

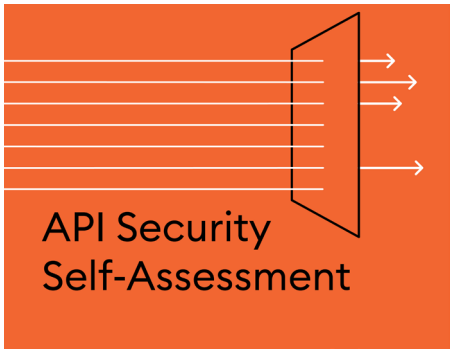
How do I get started?

That is, after all, the right question. Even though it's becoming increasingly clear that the need for API security is dire, there isn't much available that discusses it and even less practical advice on where to begin. We'll rectify some of that with the steps we've outlined below.

Where to Begin with API Security

ENABLE YOUR TEAM TO PROTECT YOUR APIS

Securing your APIs starts with connecting and enabling the teams entrusted with the heavy lifting. That begins with conversations that include your infosec and API group, preferably together.



The majority of API developers and architects don't feel that they have the expertise to secure these services properly. In contrast, infosec feels left in the dark about what APIs exist – you can't secure what you don't know about. So, building a bridge between these two groups is essential.

This may be a time to be innovative and make the commitment to securing your APIs one step further – adding security experts directly to your API development team. Embedding an infosec resource into the team will lower any barriers the group has to open communication.

If, up until this point, your security team has been segregated from the API developers, holding one (or several) knowledge transfer sessions is in order. This will get the conversation going on what can and should be done today and going forward to protect your services.

It is also extremely important to differentiate between an ongoing attack which requires minimizing the breach by quickly securing your APIs versus having the luxury of time to be able to run through the steps outlined in this section. You can take these steps on your own, but if your organization has an active API breach, the best route would be to exercise one of these options, depending on your organization's skillset:

1. Shut down the affected API(s)
2. Implement tighter API gateway security if possible. For example, implement an IP whitelisting policy that only allows a trusted IP address
3. Implement a WAF
4. Implement an ML/AI layer like Traceable or PingIntelligence
5. Engage API security experts like Big Compass

"Each layer of API security plays its part in protecting against API abuse. Each layer is required to complete the API security posture of an organization's APIs. If one layer is missing, your APIs will be vulnerable.

– Francois Lascelles,
Field CTO at Ping Identity

PERFORM AN API SECURITY ASSESSMENT

You can't lay the path forward until you have your bearings. That's why an assessment is the next step in the process of securing your APIs.

Access the Big Compass API Security Self-Assessment here.

You have a few options to get this done. You could dedicate resources and do the assessment yourself. If that's not feasible or you don't feel your team is ready to do a thorough evaluation, you could reach out to Big Compass. We're happy to help you gain clarity on where you're at and what needs to be done to reach the appropriate level of security.

You can also go above and beyond and bring in someone to do a penetration test on your APIs. This would give you the most comprehensive view of your current state. Combined with the assessment, you'll have the knowledge needed to move on to the next step.

DEVELOP AND COMMUNICATE YOUR API SECURITY VISION

With your teams communicating and full knowledge of your current API security posture, you're ready to define your API Security Vision. This plan is a marriage of the information you've gathered in the first two steps and the knowledge of your current environment and processes.

You should treat this as you would any other project. Think about:

- Costs
- Timelines
- Any other metrics you'd use for a project

Once your vision is defined, it will be time to communicate it and garner alignment with your stakeholders.

IMPLEMENT YOUR PLAN TO IMPROVE YOUR SECURITY POSTURE

It's tempting to jump to implementation, but getting API security right means laying all of the groundwork that we've just outlined first. Once that's done, only then should implementation get going.

As with the assessment, you can pull together your team to review your vision and begin putting the security changes and processes in place.

However, many companies need their teams to stay dedicated to current projects that are adding value to the business. If that's the case, you can hire API security experts, like Big Compass, to come in and review and implement the security strategies you've defined in your vision. This has the added benefit of testing your vision with a team that has seen various API security plans, both good and bad.

Gartner predicts in How to Build an Effective API Security Strategy that "By 2022, API abuses will be the most frequent attack vector resulting in data breaches". 2022 will be here before we know it so you can no longer wait to plan for API security.

We've outlined our recommendation on how to get started above, but whatever you decide to do as a next step, it should involve a plan to protect your APIs.

Chapter 7

Conclusion

We hope that this eBook has helped solidify the importance of securing your APIs. While we've walked through the benefits, best practices, and even the roles and responsibilities to accomplish this, we understand that you may still have questions.

Big Compass is here to help. We've worked with a broad profile of organizations to help them plan and create a framework for securing their APIs embedded in the API development lifecycle. From startups to Fortune 100 companies, Big Compass has the expertise to build security and governance into your API process.

If you're interested in learning more about how Big Compass can help you, or even if you just have questions about API security, send us an email, call or attend one of our webinars or Meetup events. We're always happy to talk about API security and share our extensive experience so that you can effectively protect your data and systems while still taking full advantage of your APIs.

About the Author: Aaron Lieberman

Aaron's passion for technology and for enriching connectivity between people and between systems drives him to find innovative ways to help advance organizations through technology. Aaron is the Cloud Practice Manager and an Architect at Big Compass. He has rich experience in a variety of integration environments. He brings a unique integration background where he has worked with multiple technologies to deliver creative implementations in the cloud, where most of the implementations he works on involve creating APIs and securing them. Aaron has worked with some of the industry's leading security experts to learn, adopt, and create API security best practices.

Aaron has led various implementations as a developer, architect, and development manager, so he brings the perspective of each role to every project to align people around a common goal. Aaron is also very involved in the integration community where he leads two Meetups: Denver MuleSoft Meetup and All Things Integration. He uses these Meetups to bring people together to create a community, share knowledge, and enhance collaboration. This platform allows him to engage the local community and beyond to spread knowledge and thought leadership while connecting people and ideas.

ABOUT BIG COMPASS

Big Compass builds connections that help the world get information where it needs to be, to do what it needs to do, to maximize value. We deliver elegant integration solutions for the toughest and most complex integration problems. We do this by understanding our customer's goals, thoughtfully listening, challenging ideas if necessary, and imagining the future. We then get things done in the most straightforward way, without a lot of fuss.

Big Compass is centered on one idea: Connections; system-to-system, company-to-company, process-to-process, and, often overlooked, person-to-person. We understand that nothing can be integrated without connections.

BIG COMPASS

bigcompass.com