DARKReading

4/13/2021
01:00 PM

Ryan Nolette
Commentary
Connect Directly

0 COMMENTS
COMMENT NOW

Tweet Share

5 Objectives for Establishing an API-First Security Strategy

With APIs predicted to be the most common attack vector by 2022, an API-first security strategy is critical now more than ever.

Application programming interfaces (APIs) are at the center of just about everything in the digital world. For consumers, APIs underpin everything from smartphone apps to electronic payments and more. For enterprises, APIs have enabled the shift from inefficient monolithic application architectures to more agile and modular microservices architectures that are fast becoming the standard for emerging technologies.

It's fair to say that almost every piece of software built today either uses an API or is an API. And our reliance on these crucial pieces of code to share information between applications and systems has only grown with the increased use of remote services and applications due to the pandemic.

When thinking about how APIs enable remote work, look at communication tools like Slack that depend on APIs for many features and integrations. Even a task as simple as scheduling a videoconference with a colleague is made possible by API-driven services. As APIs continually become more convenient for those with less technical training, the knowledge barrier to entry will keep getting lower.

As APIs' popularity rises, so does their prevalence as an attack vector for cybercriminals because bad actors have always loved the most target-rich technologies. When each API request is another opportunity for hackers to exploit, API security is crucial.

Related Content:

Prioritizing Application & API Security After the COVID Cloud Rush

Special Report: How Data Breaches Affect the Enterprise

New From *The Edge*: 9 Modern-Day Best Practices for Log Management

Gartner forecasts that APIs will become the most common attack vector by next year. Yet despite higher awareness of the need for API security, breaches continue to happen.

Given this context, the time has come for organizations to adopt an API-first security strategy. Companies should be mindful of the enormous — and ever-increasing — amount of data they're transmitting through APIs and make more robust security a priority at each stage in the API development life cycle.

What does an API-first security strategy look like? Here are five observations:

> **1. High visibility is crucial.** An API-first approach is all about acknowledging the API as a first-class citizen in an application's design. Given the increase in vital work that the API does in communicating between applications, APIs must have the same scrutiny of access controls that a superuser (e.g., an IT administrative specialist with unlimited privileges) would.
>
> That means an emphasis on visibility and accountability. If you can't see it, you can't account for it, and thus more visibility controls are needed for API security. The good news is organizations don't need to reinvent the wheel to get started. Web application firewalls (WAFs), OWASP vulnerability scanning tools, and smart use of Transport Layer Security (TLS) and proper authentication can all provide a consistent set of controls and visibility.
>
> **2. REST APIs are a growing target.** REST (REpresentational State Transfer) is the duct tape of technology — it defines how systems can be connected to (and interact with) each other by using HTTP requests to access and use data. REST API usage has become so widespread in enterprise application development that many companies have difficulties defining a clear picture of all their deployments. These visibility gaps make APIs harder to protect.
>
> Organizations must improve at keeping an accurate, up-to-date inventory of their REST use cases and employ secure channels like TLS and authentication keys to reduce risk.
>
> **3. Encryption of *all* data is key.** This is true not just when data is at rest, but also in transit. In this encryption scenario, the API would use TLS and authorization tokens to transmit data securely, and the data that the API is accessing should also be encrypted. Remember that this encryption strategy is defense in depth; there's a classic military saying about the crucial need for redundancy that summarizes this idea well: "Two is one and one is none."
>
> **4. Credential stuffing is still a huge problem and an evolving threat.** Credential stuffing is the practice of using an automated injection of stolen credentials to gain unauthorized access. Companies have gotten better at securing their front-end applications and webpages to defend against credential stuffing. Still, hackers increasingly have been targeting back-end APIs that historically tended to have fewer implemented security controls.
>
> Evidence of this can be seen in a recent Akamai report that showed that one in every five attempts to gain unauthorized access to user accounts is now done through APIs rather than user-facing login pages. Companies must account for this trend in their API-first security strategy and do a better job of protecting back-end APIs from these attacks.
>
> **5. Automated checks should be standard practice.** I'm repeatedly frustrated by how rarely I see automated security checks as part of a CI/CD pipeline, if they are implemented at all. A mature application security team should work with the engineering squads to design and incorporate security into pipelines and allow an organization to scale security with its product offerings.
>
> Rather than asking why this practice isn't common, let's ask why it isn't more straightforward and cheaper to implement security in the CI/CD pipeline. Why do some developer tools not include features to make security checks more manageable? Why is it so expensive to audit for API

security issues on running services? What are the dev tool vendors doing to improve this experience for their users and help developers punch above their security weight class?

I hope and expect the API industry will step up and better address these questions and concerns in 2021 and beyond.

As these five points show, pursuing an API-first security strategy and baking API security into the software development life cycle requires prioritization and intent. But it's well worth the effort. A few minutes of proactive security testing can save hours or days of downtime and legal fees down the road.

Ryan is Postman's Technical Security Lead and co-author of AWS Detective. He has previously held a variety of roles, including threat research, incident response consulting, and every level of security operations. With over a decade in the infosec field, Ryan has been on the ... View Full Bio

Recommended Reading:

COMMENT | EMAIL THIS | PRINT | RSS