fastly

WHITEPAPER

The AppSec Guide for Multi-Layer Security

8 tactics to get you on the road to a unified security strategy

This guide will help you build more security into your CI/CD workflows, reduce maintenance, shrink your attack surface, and save money. Improve your security posture with tactics for the different layers of your organization and architecture.

Introduction

The technology in use by organizations continues to grow more sophisticated over time, and as that happens, the attacks used to try to exploit it are getting more sophisticated and harder to manage as well.

A recent IDC survey on DevSecOps confirmed that implementing security "across multiple cloud environments is both the biggest risk addressed by DevSecOps and the biggest technical challenge."[1] But that's just one of many challenges facing organizations. It also remains difficult to hire and scale security experts for internal teams, and budgets are stagnating or shrinking.

If you're one of the many people trying to manage an expanding scope of security work with the same or shrinking resources, then it's time to adopt the modern security tactics that get you to a better security posture while making life easier – saving time, money, and headaches for your team.

---

1 IDC, DevSecOps Adoption, Techniques, and Tools Survey, 2023, Doc #US50137623, May 2023

Table of Contents

What do we mean by multi-layer security?

The short version is that relying on one single point of protection like a web application firewall (WAF) is less effective than addressing security at multiple layers of your organization and infrastructure. It can be thought of in a variety of ways. For example, WAFs protect you at layer 7 of the OSI Model while network-level mitigation occurs at layers 3 and 4. But you can also think about the boundaries between layers of an organization's tech stack and the boundaries between modules in a complex application. No matter how you look at it, you will be safer and stronger, and spend less time handling emergencies if you build security across your organization at multiple layers, protecting as many boundaries as possible.

A good WAF can stop malicious requests, but you get more protection by identifying and blocking malicious requests before they reach your WAF, adding an extra layer of security to your organization's architecture. Similarly, if your code is running in an environment with strong isolation and sandboxing features, then even if a malicious request gets past your WAF, even a successfully executed exploit can be contained to have very little impact.

Choosing the right content delivery network (CDN) has a bigger security impact than you might think. People know Fastly's CDN offerings for its performance superiority, but our customers also get significant security benefits just from moving to our delivery network. A Fastly commissioned study by Forrester Consulting recently noted that "Security incidents decreased for representatives' companies as they spent less time troubleshooting customers' issues. One company saw a 40% decrease in security incidents with the deployment of Fastly CDN." Shifting traffic to a CDN that reduces security incidents is another way to tackle some of your security issues at a different layer.

The following security tactics live across many different layers of an organization. Applying any of them individually is good, but applying them together will make an even bigger difference.

What do we mean by attack surface?

An attack surface is a way to talk about the total number and parts of an organization's technology footprint that represent potential points of attack. A smaller attack surface is better – it means there are fewer points that can be attacked and exploited, but it also means that the teams responsible for keeping these points secure have an easier job. There are lots of factors that can expand or shrink an organization's attack surface, including making sure all applications and APIs are protected by a WAF, deprecating high-risk pieces of the tech stack, implementing best practices, and reducing complexity so that the remaining attack surface is easier to protect.

It can be very effective to combine a multi-layer approach that mitigates security threats in different layers of an organization's technology with an approach that continually seeks to minimize the total attack surface. The following tactics can help an organization improve its posture in exactly these ways.

Tactic 1: Smarter, modern threshold blocking

When legacy WAFs talk about threshold blocking it's in very simplistic terms. They can do basic rate limiting based on rates alone OR they can make all-or-nothing decisions on the content of a request being caught by one of their rules. The smarter, modern version of threshold blocking makes it easy to set smarter blocking conditions.

Modern threshold blocking in a Next-Gen WAF accomplishes two things:

1. Enables decision-making based on the combined intelligence of both the content of the request AND a threshold like the pattern or rate of requests around it (or source IP or other factors). Setting rules that take into account both the request content and rate provides significantly more flexibility and nuance, which all leads to higher accuracy.

2. It allows you to separate blocking decisions from initial detections.

This is a great way to reduce false positives by making it easy to set rules that let through initial requests that are suspicious, but clearly identifiable as attack attempts until the rate of them makes it clear that it's malicious. The ability to apply time-based thresholds along with rules about request contents gives you more control and the ability to easily build rules that are more effective and less fragile than a legacy approach of immediately blocking any incoming request that matches a regular expression (regex) rule.

This is the difference between making an informed blocking decision vs. an uninformed, all-or-nothing and overly blunt blocking approach. Without smarter threshold blocking you're limited to applying strategies of indeterminate, uninformed blocking that put you in a no-win situation to either generate more false positives or block more healthy traffic.

Easy threshold blocking with Fastly

First of all, and importantly, Fastly's Next-Gen WAF (NGWAF) makes it easy for you to do smart threshold blocking. It is outright impossible (or at best, significantly more difficult) to do with legacy WAFs.

Once you're able to do smart threshold blocking, you can shift your blocking strategy. You can minimize false positives by allowing traffic through at first, and then quickly and intelligently block once there is a higher confidence that the traffic is malicious. You get the best of both worlds – lower false positives with significantly better security.

Bonus! Easy migrations! The NGWAF comes with out-of-the-box thresholds in place that are very conservative (50 attacks-per-minute), and this makes it easy to turn on blocking mode right away with confidence that you're blocking real attacks (like all of the OWASP top ten and other advanced attacks) without accidentally stopping anything legitimate. You can tune down the thresholds over time as you observe that it's functioning as expected, add some custom rules that you may want, and gain confidence while enjoying the improved security from the start (or you always have the option to jump straight to instant-blocking). Migrations with legacy WAFs can involve months of running in logging mode and then poring over the logs before ever blocking a single attack, and that's just the beginning of tuning and maintenance effort that you simply don't have to deal with if you select a modern WAF that supports things like easy threshold blocking.

Tactic 2: Eliminate Regex inaccuracy and maintenance

Regular expressions (regex) are a powerful tool to identify something when you know exactly what it's going to look like, and exactly what it's not going to look like. The hitch with modern attacks and exploits is that they try myriad approaches until one bypasses defenses. And then your problem with regex multiplies because now you have to write a new rule to patch the hole (assuming you catch the exploit), and then deal with rule tuning's unintended aftermath.

It's a common occurence that a regex rule accidentally blocks valid traffic. The maintenance demands of continued tuning for each new instance and code deployment become unmanageable. You have to decide whether to leave the rule in place and block good traffic (yielding bad user experiences), or turn the rule off and leave yourself with vulnerabilities in your WAF.

Simply put, regex is inadequate for modern security rule-building and maintenance and leaves organizations less secure over time. There are better ways to do this.

Easy rule building and maintenance with Fastly

Fastly's NGWAF uses SmartParse because it's smarter, significantly more accurate, easier to use, and easier to maintain than regex. SmartParse is context-aware and understands the syntax and components of OWASP top ten attacks (and many others), which means it can tell the difference between a genuine attack, compared to seemingly similar patterns that are actually legitimate user inputs. For example, legacy WAFs often block requests containing words like "drop" or "tables" somewhere within the request for perfectly normal reasons, and not in a location that poses any risk. This causes false positives. With regex you would have to write exclusion rules anywhere in your application this false positive has appeared, or may appear in the future. And then you have to maintain it over time as the application is updated or you find new edge cases. This also opens up potential security vulnerabilities as writing exclusion rules against a rule that was meant to stop attack payloads can create more unknown openings where your application is vulnerable.

SmartParse is so accurate that about 90% of Fastly NGWAF users enable full blocking mode, in production environments. They trust SmartParse to block real attacks while permitting legitimate traffic, and because they can set up immediate blocking of identified attacks like SQL injections while employing threshold blocking (from Tactic 1) for any requests they want to react to more conservatively.

Creating custom business logic rules in the Fastly dashboard is incredibly fast, simple, easy to maintain, and human-readable, and because you're building alongside the detection accuracy that SmartParse provides, it is powerful without requiring endless maintenance and convoluted exclusions – and you never have to translate anything to or from regex again. As one example, this docs page details how simple it is to build advanced rate limiting rules for the NGWAF. You can see how it removes the complexity and constant maintenance of regex and makes effective SecOps practices accessible to more people on your team – not just the regex and security wizards. This is part of the reason that we see many customers move some or all of their NGWAF management into other parts of the org, and not just their already overloaded SecOps teams. We see DevOps, application developers, and engineers from all parts of an organization successfully managing their WAF instances and improving their overall security posture.

Tactic 3: Preemptive and threat-informed IP blocking

IP address reputation is a powerful tool for blocking traffic, but only if you have a high degree of certainty in the reputation score, with the address confirmed as sending malicious traffic. Ensuring you don't block legitimate traffic demands accurate threat intelligence around IP reputation, which can be complicated to collect and even harder to make actionable. Shared IP spaces add another complication because blocking an IP address for a shared IP space like a hotel, coffee shop, or conference, might cut off a lot of good traffic because of the actions of one malicious actor.

If you can reliably take action accurately based on IP addresses, you can block a significant volume of malicious requests before they even reach your network, However, this requires three things that most vendors cannot provide – extremely accurate IP threat intelligence, threshold-based blocking, and simple, powerful rule-building.

Easy, accurate threat-informed IP blocking with Fastly

Fastly's Network Learning Exchange (NLX) is a highly accurate source for IP reputation intelligence. The data is collected from over 90,000 apps and APIs that inspect over 4.1 trillion requests per month[2]. But it's not just about the volume, it's about the accuracy. NLX's accuracy in IP reputation comes directly from SmartParse's accuracy in threat detection – it has such low false positive rates that NLX represents a list of IP addresses, updated in real time, that is known to be the source of confirmed malicious attacks.

Armed with NLX data that is built directly into Fastly's Next-Gen WAF, the threshold-based blocking from Tactic 1, and the easy rule-building from Tactic 2, it's easy to create rules for NLX-signaled IP addresses. You could block them outright, or combine them with other anomaly signals allowing you to make more informed decisions when handling suspicious requests. For example, to ensure that valid traffic from a shared IP space is still allowed through until you know there's a threat, you could easily set a low threshold for requests before blocking any NLX-signaled IP address that also sends a malicious request. As NLX is updated in real-time, the rules you set are automatically applied to new IP addresses that have started to issue attacks, and automatically removed from IP addresses that are no longer identified as threats.

NLX helps you react more aggressively to traffic from IP addresses that are known to be current or recent sources of malicious traffic, while thresholding allows you to still err on the side of allowing traffic through until you know it's a threat.

2 Trailing 6 month average as of June 30, 2023

Tactic 4: Treat your content cache as a security layer

Cache hit ratio (CHR) is a core metric of any system that includes a cache. It's measured by observing checks for content that may be in the cache. We calculate cache hit ratio by dividing the number of cache hits by the total number of cache checks. If each check can result in either a hit (content found) or a miss (content not found), then the ratio is:

$$CHR = \frac{hits}{hits+misses}$$

Historically the CHR has been understood as a performance metric because the more you cache, the more money you save and the faster your site performs. When you increase your percentage of hits-to-misses you get these benefits:

1. Reduced hardware capacity needed at origin
2. Lowered architectural complexity needed at origin
3. Significant reductions / savings in egress charges
4. Latency reductions from serving content from cache

But it's not just about performance. There are also serious security benefits that come with improving your CHR which have often received less attention, and if you start thinking about CHR as a security metric as well as a performance metric you can get that extra protection.

Increased risk often comes from increased hardware management, increased application complexity, increased architectural and deployment complexity, and contending with systems that are harder to reason about. An increase in your CHR is an indicator that you are successfully reducing the amount of hardware you need, decreasing application complexity, decreasing architectural and deployment complexity with simpler systems that are easier to reason about, and guaranteeing more uptime and reliability via a hardened and automatically responsive CDN partner.

CHR isn't a direct measure of security, but improvements you make in your CHR are evidence of succeeding at the things that help you create a more secure system.

Easy CHR improvement with Fastly

Different types of applications have different upper limits for their CHR – some tougher situations might be hard pressed to exceed 80% or 90%, but many Fastly customers are able to achieve 95% or better. This is as big a win for security as it is for performance or cost savings.

With Fastly you can increase your CHR to the point that it functions as an origin offload that shrinks the attack surface of your origin by significantly reducing the request volume to origin. Here are 6 ways to cache more with Fastly – many of which are not possible with other networks.

1. **Increase your TTLs so content is eligible for eviction later**

   The main parameter you have that controls the cacheable lifetime of an object is the TTL, or Time to Live. This dictates how long Fastly can reuse that content to serve future requests. A shorter TTL will lead to your content being evicted sooner, resulting in cache misses and more fetches to your origin.

   If you think your content is too dynamic or changes too often to use a long TTL, I can almost guarantee you are wrong (and you should schedule some time to talk with us). One of the tools we provide to

support dynamic content is Instant Purge, so let's talk about that next.

2. **Use long TTLs and evict content when it changes with Instant Purge**

   Fastly's Instant Purge API allows you to evict content from our network, on demand and extremely quickly: this takes an average of 150ms to execute a purge worldwide. With a little integration work in your application, you can take advantage of very long TTLs (how about a year?) even with rapidly changing content, like API responses or live sports scores. You may find that you can cache a lot more content than you think.

3. **Enable Origin Shielding**

   Fastly's Origin Shielding feature allows you to designate a specific POP to act as your origin for our other POPs; this can significantly increase the likelihood of finding your content in cache, with a corresponding reduction of load to your origin. Read more about origin shielding →

4. **Temporarily serve stale content**

   Configuring your Fastly service to serve stale content keeps your service available even if your origin is unavailable.

5. **Choose a CDN with fewer, more powerful POPs with more storage**

   This sounds a little counterintuitive, but the fewer POPs a CDN has, the more likely it is for any single POP to already have your content in cache. The tradeoff, of course, is that each of those POPs has to have much more storage capacity in order to serve a larger combined content pool. This is one of the core principles of Fastly's CDN architecture. Read more about the benefits of modern POPs →

6. **Move image handling to the edge**

   Enabling Fastly's Image Optimizer allows you to move image logic off your origin, performing image transforms and optimization at the edge. It also has very tight integration with Fastly's cache, ensuring that even for a wide variety of device-specific images, we make maximum use of available storage.

   It's integrated with Instant Purge, as well: purging your high-resolution originals also purges the images that have been derived from them. You can still take advantage of long TTLs, even with changing content.

This is just a taste of the ways in which Fastly was built to cache the uncacheable. Read more about it in this blog post: Understanding CHR as a security metric.

Tactic 5: Secure more of your logic on the serverless edge

Caching, and even advanced caching is just the start. There's a lot more that should be moved off of your origin and onto the edge in order to shrink your attack surface, reduce the complexity you have to manage at origin, and outsource more to the edge. You even get performance gains by locating more of your logic closer to your users so that they get lower latency and better response times. There's only one catch – you have to choose a serverless edge environment that is secure by design and provides the level of security necessary to deliver these benefits.

For example, if your HTTP redirect service is served from a secure edge environment then attacks on your HTTP redirect service are no longer a concern for you.

Easy, secure logic at the edge with Fastly

Fastly's Compute@Edge environment is more secure and more performant than the competition. Compute@Edge delivers per-request sandboxing and isolation, which means every request that occurs in the environment is fully isolated from every other request, so you get a very high level of security for everything done on the platform by default, as well as built-in mitigation for the potential impact of any exploit that could ever be achieved because the attacker only gets access to the data within that one request and nothing else.

Every piece of logic that is moved off of origin and onto Compute@Edge becomes a huge security win by shrinking your attack surface and management overhead at origin, while running more of your operations in a more secure environment. You can do just about anything you want using Compute@Edge – check out the Modern Application Development Playbook. Some of the most common applications people start with are authentication, geolocation, personalization, and SEO.

Bonus! Bundled services like Image Optimizer!
If you want more logic at the edge you can also take advantage of some pre-built services like Fastly's Image Optimizer that lets you offload all image transformation and optimization to the edge and deprecate your own internal image services for image management. Learn more here or check out the Image Optimization ebook for more details.

Tactic 6: Use a network that is secure-by-design

Not all networks are created equal, and it's not just about performance, feature sets, transparency, and configurability. Serving your content over a modern network architecture can have significant security benefits as well. It may not be what you thought you'd find in a security tactics white paper, but moving onto a network that helps your organization's networking benefit from secure-by-design architecture translates directly into a reduction of security incidents and escalations.

Easy network and platform security with Fastly

Fastly's modern network and architecture provide benefits like rapid traffic inspection to monitor for DDoS and bot attacks. Massive computing power allows the inspection of data packets as they flow through the network, including SSL traffic. The Fastly network is built to quickly identify malicious code and attacks, and immediately protect customers by automatically rerouting or blackholing (dropping) the bad traffic. Fastly sees 1.4 trillion requests per day[3] through its content delivery network, and that amount of information enables advanced threat intelligence at a global scale.

Fastly takes network security so seriously that it built its own TLS certificate authority (CA) called Certainly. It's a lot of work to make a new CA, but there are some important benefits. By default Certainly issues 30-day certificates for all Fastly customers at no extra cost to ensure that any successful exploit has limited impact. It also means that if large-scale certificate issues arise, even globally, Fastly customers are not stuck in a queue behind millions of other requests when everyone is scrambling to renew their certificates. Lastly, it also means that Fastly customer support is a one-stop shop for all networking needs – it doesn't matter if the problem is with the certificate, the network, the configuration, or anything else you might have an issue with. Fastly's customer support has the power to address and resolve everything from end-to-end.

But don't take our word for it. Fastly commissioned a Forrester Consulting Total Economic Impact™ (TEI) study examining Fastly's Network Services, and it highlighted significant security benefits just from moving your traffic onto Fastly's network. See the next page for some of the findings.

3 As of January 2022

Forrester TEI study findings

Improved networking security

- Security incidents decreased for representatives' companies as they spent less time troubleshooting issues. One company saw a 40% decrease in security incidents with the deployment of Fastly's CDN.
- With Fastly CDN's core capabilities, interviewees shared how their companies were able to reduce cyberattacks and malicious traffic with layered security features that would kick in automatically. A chief security officer in e-commerce discussed: "We turned Fastly on, and suddenly all the malicious stuff, all the basic application-level attacks that used to trigger our operations center and response on our side, were just automatically dealt with. [It's] hard to quantify cost or value on that, but I can sleep better."

Ease of use of Fastly's Next-Gen WAF

- A chief security officer in e-commerce described the impact for their company: "Another part of the economic impact is the web application firewall. Imagine all your traffic going to the edge: You check a box in Fastly, and it detects and removes attacks."
- "Prior to us enabling WAF as part of the Fastly [platform], our engineers were getting two or three escalations per week during nonbusiness hours. After they implemented the security tools from Fastly, that went to nearly zero." – Chief security officer, e-commerce

Improved site reliability

- Interviewees shared that their companies were able to decrease downtime on their websites based on Fastly's capabilities to stop malicious activity and quickly react to traffic surges. A director of engineering in travel and hospitality shared: "Fastly, in multiple situations, blocked large swaths of traffic hitting us, which then in turn prevented us from going down because we couldn't have coped with that sort of scale-up. Those incidents could have lasted several hours where we were trying to scale back up for that traffic surge."

Vendor consolidation and cost savings

- A chief security officer in e-commerce shared that their company saw savings for eliminated current point solutions equal to their entire investment in Fastly.

Read the full TEI study

Tactic 7: Extend your WAF without point solutions

There are always areas of your attack surface that aren't covered by default, yet trying to cover them all with one-off point solutions or add-ons can add so much complexity, maintenance overhead, or added cost that it outweighs the protection that is being offered. Legacy WAFs are harder to extend in ways beyond their design – and even if you can, they bring the same limitations about blunt rule enforcement that got us here in the first place.

Deployment is its own challenge for SecOps teams who want to provide tools to their organization. Deployment solutions are notorious for breaking or for their fragility in their relationship to the CI/CD workflows used by the engineers in the rest of the organization, which can result in broken security solutions at best, or at worst, a cascading impact into broken applications with bad user experiences.

Easy WAF extensions with Fastly

Fastly's Next-Gen WAF already supports integrations with top vendors for CAPTCHA, identity and access management, and bot mitigation. It's easy to plug data sources into the NGWAF and build business logic with its rule-builder interface. But importantly, it lives on a global edge network, which delivers more benefits than just performance.

The edge serverless computing environment offered by Compute@Edge delivers all of the security benefits discussed above in Tactic 5, but also makes safe, fast, global deployment easy. Building in the Compute@Edge environment is perfect for SecOps teams who want to create solutions for their organizations and know that they will be safe and reliable to deploy with the NGWAF in the workflows that their organization's developers are already using.

Imagine using a bit of JavaScript to write some logic, and that's all it takes to safely extend your WAF for login discovery, malware verification, or password validation – all on the edge, all reliable, and always safely deployed along with the WAF in an application developer's existing CI/CD pipeline. Working on the edge can speed up security tool development for an org just as much as it speeds up application development.

Tactic 8: Better CI/CD integration improves security

Engineers and DevOps are happy when things integrate into their existing workflows, but when it also improves or maintains your security posture it's a reason for SecOps to get happy as well. On the networking and content side, if you use a secure-by-default network with built-in security benefits, then it's easy for engineers to provision their applications through normal workflows and retain those integrated security benefits.

It can be even more important for WAFs because it helps an organization ensure that a proper WAF deployment is baked into their CI/CD processes so that everything is protected, even as changes and updates are made, or as new applications, services, and APIs are spun up. It also speeds up your software delivery and innovation when rigid or limiting SecOps requirements aren't creating bottlenecks.

CISOs are understanding that they can't slow down the engineers, and their policies need to enable their organization rather than block it. But modern organizations using CI/CD workflows make frequent updates and changes, and many legacy tools simply can't adapt. The tuning and maintenance required for regex rules as application updates are deployed tens, or hundreds of times per day can become overwhelming and create pressure for more leniency in security rules. How often do new features introduce new false positives? Regex rules often broadly match criteria, so something that used to be blocked because it wasn't expected before might now be expected in the new feature. You didn't previously expect to see address data, but maybe a new feature introduces an address input field. It's impossible to tune and monitor and test effectively at that rate of change, and so the choice becomes either building rules that are too narrow and exclusions that are too broad, or repeatedly creating bad user experiences and constant scrambles to fix things as new features break unexpectedly.

Easy integration of security into CI/CD with Fastly

Fastly believes in easier DevOps that enables better SecOps, and we've built our NGWAF to deliver on that in four different ways. First, you get one WAF that can be deployed anywhere, across an entire organization – at the edge, on-prem, in the cloud, multi-cloud and hybrid. Not enough? You can also deploy in containers, service meshes, API gateways, reverse proxies, ARM environments, and AWS Lambda. It's all managed with a single dashboard, and there's no lock-in concern with multi-CDN support. When a single, more effective WAF wraps every app, API, and service for an organization it represents a huge reduction in complexity and a win for sensible security practices.

The NGWAF also requires less tuning than the competition. Many of the tactics mentioned earlier in this paper, like SmartParse, IP reputation from NLX, and smart thresholds create less fragile protection. Frequent deployments from a CI/CD workflow don't represent challenges because you're not breaking regex rules or scrambling to write exclusions.

Consolidating to one WAF isn't just about simplifying deployment and billing. Every app and API behind one WAF means all of your reporting and logging is in one place and under one dashboard. The NGWAF's accuracy reduces alert fatigue, but it also reduces vendor, dashboard, and permissions fatigue.

Easy management and rule-building let developers self-manage effectively without having to know the ins and outs of regex wizardry. It is more common to see right-shift organizations empower their application developers with responsibilities that used to live only in SecOps as they find tools like the NGWAF that are easy and effective to use.

Conclusion

The future of comprehensive application and API security rests in a few areas. First, smarter tools that require less effort and overhead to manage. Second, better integration into the processes already in place within an organization so that productivity is accelerated rather than bottlenecked by security. And third, a shift from cobbling together point solutions from many different vendors to cover every security risk separately to using platforms that can consolidate many layers of security practices under a single, more efficient dashboard, invoice, and partnership. If you want to learn more about how easy a migration to this kind of future with fewer headaches could be, get in touch with Fastly today.

Additional Resources

- Cache hit ratio as a security metric: Take a deeper dive into understanding its security impacts, and how to improve your CHR. Read the blog post →
- Preemptive IP threat intelligence: Learn more about the Network Learning Exchange (NLX) and the smarter way to do IP threat intelligence. Read more →
- Building applications on the edge: Learn about the performance and security benefits of a secure-by-design platform. Get The Modern Application Development Playbook. Download now →
- Regex in Retrograde: Read Kelly Shortridge's explanation of why regex isn't right for security. Read the blog post →
- The DevOps Roadmap for Security: Learn more about unifying the efforts and fostering collaboration between your Security and DevOps teams. Download the ebook →