Cybersecurity on Netlify: Unlocking the Jamstack for scalable applications

Cybersecurity on the Jamstack

Millions of developers, thousands of businesses, and hundreds of leading enterprises use Netlify to build, host, and deploy websites. We take our responsibility to ensure the security of those sites seriously. Netlify employs best-in-class practices to provide a 99.99% uptime SLA for our Enterprise customers, including core capabilities, such as:

- **Encryption:** All traffic over our networks is TLS encrypted and all sensitive information like access tokens, SSH deploy keys for Git providers or HTTPS private keys are encrypted at rest.
- **Data center security:** Netlify uses globally-distributed data center partners that comply with leading security policies and frameworks, including SSAE 16, SOC framework, ISO 27001 and PCI DSS. All our data center partners follow industry best practices and comply with a wide array of standards. Netlify reviews compliance with these policies regularly.
- **Identity and Access Management:** Teams can sign in to the Netlify App with G Suite, Okta, OneLogin, Ping Identity, or most identity providers that support SAML 2.0. For teams on Netlify's Business or Enterprise plans, Netlify supports integrating an existing SSO provider to authenticate users.
- **Penetration testing:** Netlify regularly performs third-party penetration tests and engages the wider security community. The ability to run such tests or view reports are reserved for customers on Netlify's Enterprise plan.

Additionally, there's one important element of security that's baked into Netlify's architecture: the Jamstack itself.

Table of Contents

Cybersecurity in a Jamstack World

Modern web applications—where the Frontend is decoupled from the Backend and deployed to the edge—unlock a completely new approach to web security that's less prone to attack and far less complex to build and manage. It's time to rethink security and ask the following questions:

- Where and when is code being executed?
- Who/what has access to the backend and databases?
- Is there an origin server? Where does it live?
- Where are the attack vectors on my sites?

Jamstack advantages

Jamstack is quickly becoming the preferred architecture for the web, in no small part because of its security advantages. Using Git workflows and modern build tools, pre-rendered content is served to the network edge and made dynamic through APIs and serverless functions. By abstracting server-side processes into microservice APIs, you gain the following:

- Minimal surface area with largely read-only hosting infrastructure
- Decoupled services exposed to the build environment and not the public
- An ecosystem of independently operated and secured external services

All of these principles improve security while also reducing the complexity we need to manage and maintain. Let us dive into these principles from a security perspective.

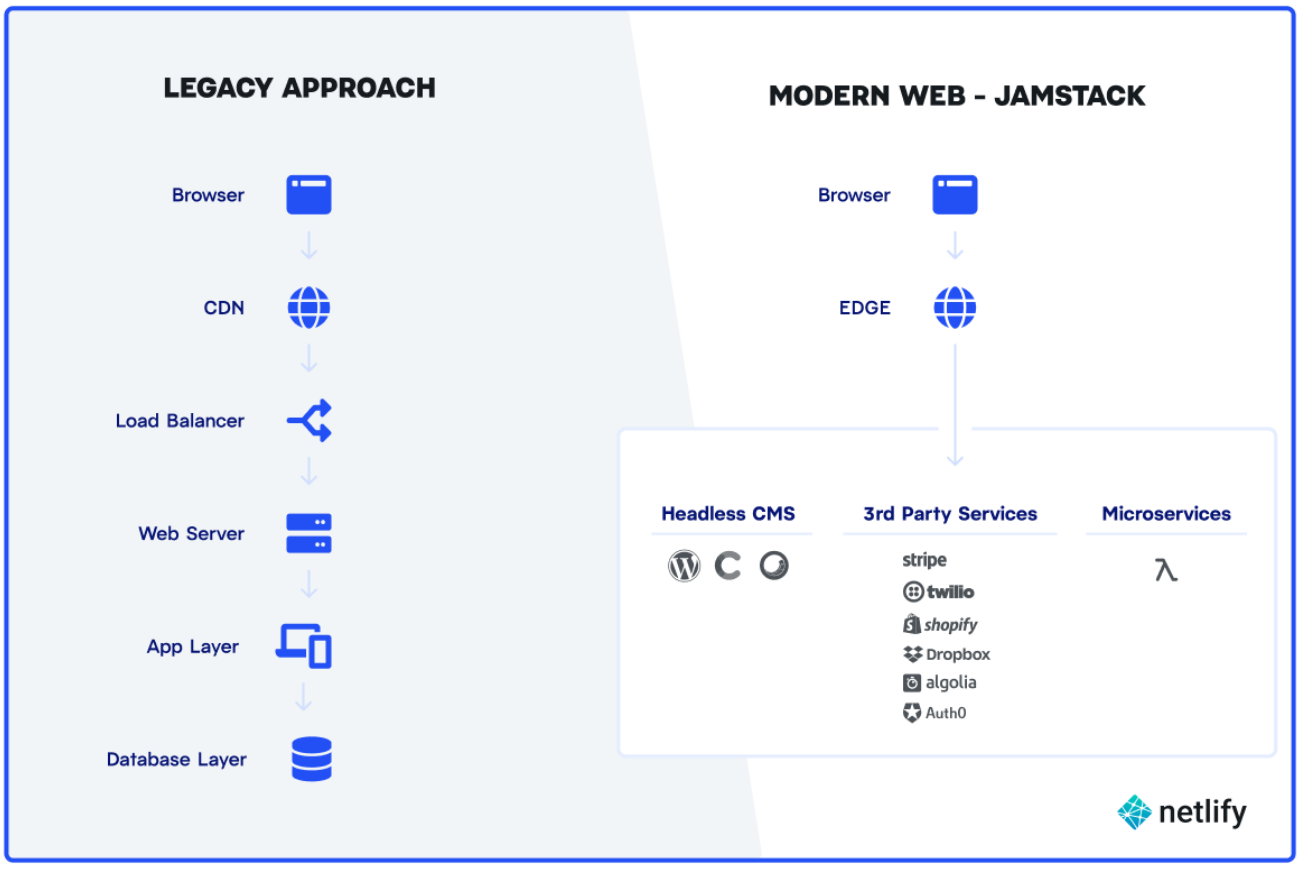
Less Vectors for Attack

The term *surface area* is often used to describe the amount of code, infrastructure, and the scale of the logical architecture at play in a system. It's the areas that are publicly exposed

for people to attack. Reducing the surface area is a good strategy toward improving security and minimizing avenues for attack.

The problem with traditional websites is that they tightly couple the frontend to the backend. Changes to any part of the site require a lot of coordination between multiple teams across the stack. But more importantly, a request made to the site will have to traverse through multiple layers of proxy, servers, databases, and backends in order to respond to the user. From a security perspective, each layer can add more opportunities for attacks. For instance, SQL injection attacks to the database or XSS attacks to your web servers.

The Jamstack benefits from a smaller, simplified stack when compared to traditional architecture as seen in the diagram below. Fewer pieces of infrastructure mean fewer attack vectors and fewer elements your team needs to spend time patching and protecting. When you host sites on Netlify, **most** of the site is statically built and deployed to the Edge as HTML, CSS and JS, really reducing the endpoints to attack.



The infrastructure responsible for serving your site does not include the logical code that's executed on Netlify's servers at request time. Application code runs on Netlify's build infrastructure (prior to deployment) and when using Netlify Functions (in production). Both environments are ephemeral; temporary environments spun up just long enough to execute each task. That means there are no idle environments to attempt to exploit, and limited exposure to public networks.

Tip: Security is a shared responsibility between Netlify and you. Be sure to follow best security practices, such as writing secure code and do not store secrets as unencrypted environment variables.

Any server-side processes are best abstracted into microservice APIs including third-party microservices for specialized services like messaging, search, and payments, and one should leverage the domain expertise of specialists behind those services.

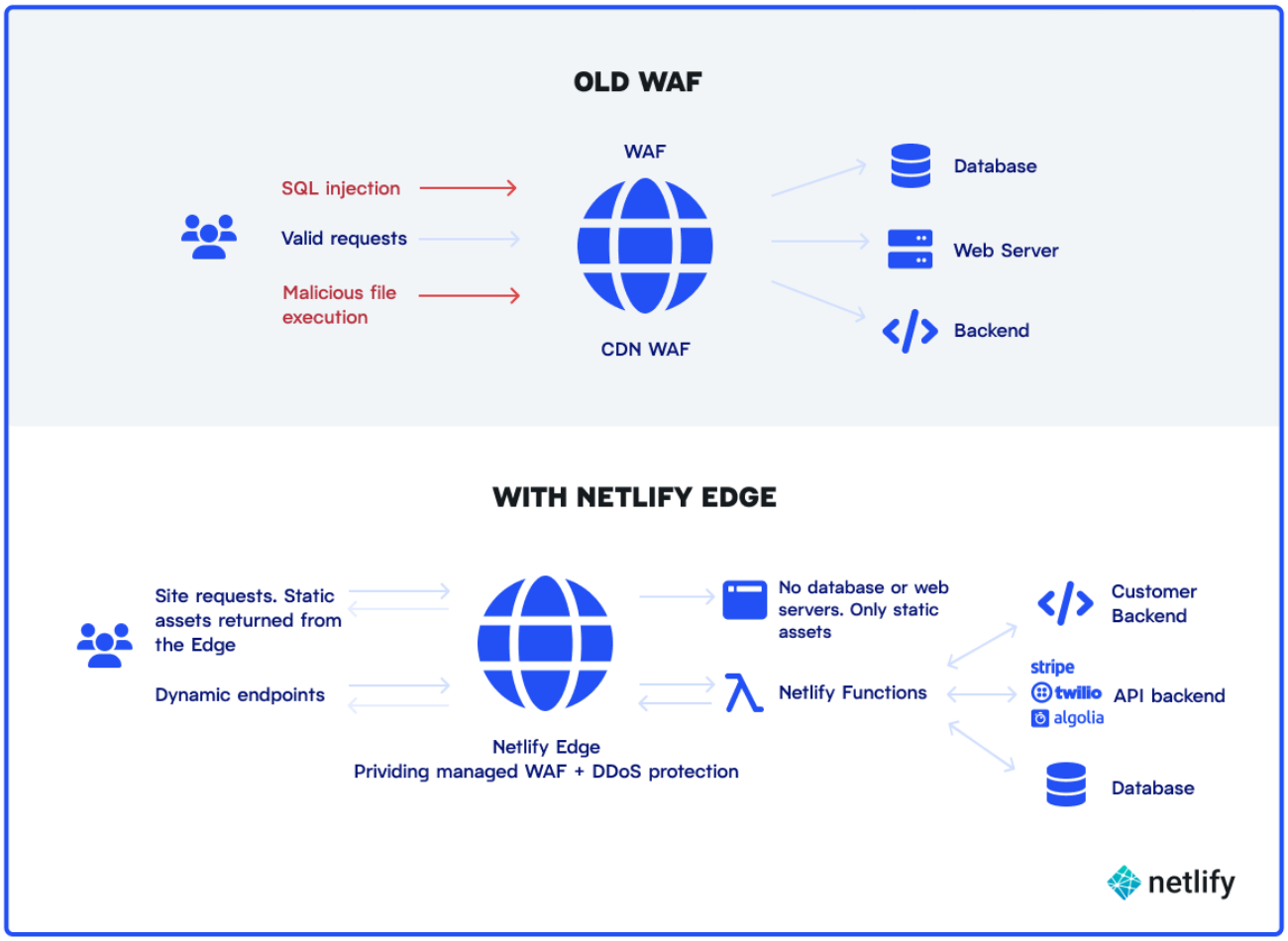
The role of the Web Application Firewalls with Netlify

WAFs are designed to protect origin web servers and endpoints.

With legacy web architecture, the site/app is hosted on a webserver and teams have to purchase a CDN with WAF capabilities to protect this server (the origin) as well as all the other endpoints like database and backends that are part of the request loop.

SQL injection attacks that target databases and malicious file execution that target web servers and try to modify content, are both common and protected against by the WAF.

The diagram below compares this legacy approach to Netlify's Edge:



The frontend of apps built using a modern Jamstack architecture is pre-built. That means the web server and dynamic endpoints like the CMS and backend are completely eliminated from the request loop. Malicious attackers have far fewer endpoints to attack and only static assets are retrieved from the CDN. Typical requests involve static assets that are served straight from the Netlify Edge - these assets already exist on the Edge and no other endpoints are hit to serve those.

For dynamic requests that hit a Netlify Function or a database as part of the application, these are proxied via the Netlify CDN which provides a managed WAF and DDoS service that will handle IP blocking, bot protection and malicious requests for the customers. For third party services that a site uses, say a database like Fauna or payment service like Stripe, we recommend you use the security postures and best practices that they provide while using their APIs.

The concept of running plugins as part of your sites also goes away with the Jamstack and Netlify, along with the need to maintain, update and secure these plugins. Netlify also offers the ability to sign any outgoing proxied request to other services so APIs can easily deny all traffic not passing through Netlify's Edge.

This model also makes for better logical separation of individual services and underlying capabilities, which can be advantageous when it comes to maintaining a suite of capabilities across your site. The result is clear separation of concerns in addition to security responsibilities.

How Netlify Manages Cyberattacks

Let's explore what this architecture means for common categories of cybersecurity attacks. Specifically, we'll need to dive into what makes Netlify Edge different from traditional CDNs, in order to understand the security benefits that are in place.

CDN designed specifically for scalable Jamstack Applications

Unlike a traditional CDN, the Netlify Edge does not sit in front of an origin server. And we add our own software logic to the standard CDN setup. This logic allows for a developer optimized setup that includes instant cache invalidation, build snapshots, instant rollbacks, and atomic deploys, which ensures your site is always in a consistent state. Our CDN edge nodes can make decisions normally made on the origin server. Requests for passwords, redirects and proxies don't bounce through server relays; they happen as close to your users as possible.

When deploying to Netlify Edge, the need for another CDN becomes obsolete. In fact, we do not recommend using another CDN in front of Netlify Edge just for the DDoS/WAF capabilities, as both are handled by Netlify Edge.

DDoS protection from Netlify

Netlify Edge has all the uptime benefits and core functionality of a regular CDN and more. Netlify operates a multi-cloud CDN for redundancy, and has different layers of DDoS mitigation protection across the cloud providers. And because Netlify utilizes multiple cloud providers, if one is attacked or suffers a shutdown we automatically route traffic to the next closest node. Our DNS based traffic director will route users directly to the closest active CDN node without any intermediate CNAME or A record. This level of abstraction allows us to deal with DDoS attacks, route around localized outages, or spin up other servers to absorb or quarantine bad traffic very quickly.

Netlify Edge serves around petabytes of traffic, running websites and web applications created by more than 2 million developers. Netlify employs both automated security mitigation at the network edge as well as manual optimization by Netlify's Site Reliability Engineering (SRE) team. As attacks become more sophisticated, Netlify's system automatically adapts. The SRE team also constantly monitors and implements new techniques to avoid future attacks and keep the network performant. Specific capabilities Netlify Edge offers to handle attacks include:

- **IP Blocking at the Edge:** Netlify protects websites against Application Layer attacks by scanning logs for misbehaving IPs and blocking them at the edge.
- **Rate controls**: Netlify controls the rate of requests against multiple endpoints that sites might hit, automatically protecting against volumetric attacks.
- **Monitoring**: Traffic patterns and endpoint traffic are monitored at all times to detect and thwart abnormalities before they affect site performance.
- **Security monitor**: Websites and web applications are actively monitored for attacks and downtime. Any material degradation in performance or downtime will immediately page Netlify's SRE team. Available with Netlify's Enterprise plan.
- **Logging**: Use Log Drains to connect site traffic logs and function logs to third-party monitoring services like Datadog for further analysis and alerting, increasing threat posture awareness. Available with Netlify's Enterprise plan.

Eliminating SQL injection attacks

Using a Jamstack architecture with Netlify, the database and the backend are hit at Build Time, where the content and data are used to create the static assets for the site. Netlify's build infrastructure is ephemeral and runs within secure environments that are torn down at the end of every build. Requests made to the site at runtime are retrieved right at the edge, with fewer calls to a database than traditional server side applications that hit a database on almost all requests. For applications that do use a database, like StepZen, Fauna, etc., these services secure the APIs and data that they host. Because of this architecture, SQL injection vulnerabilities—where attackers interfere with application queries to the database—are effectively eliminated.

World-class Site Reliability Engineering (SRE) Team

Not only is Netlify's infrastructure built with advanced security capabilities, but Netlify's SRE team acts as an extension of your team. We put in place monitors and alerts for your sites and automatically comb them for traffic pattern anomalies and spikes, effectively handling them as needed. For DDoS or similar layer 7 attacks, the Netlify team will handle the attacks straight from the edge nodes. Our DNS based traffic director will route around global network issues and potential latency, ensuring users are served from the best global node. Enterprise-level support enables companies to reach our support and platform teams in case of an issue and escalate a ticket by phone or email.

Conclusion

Netlify knows that trust is built with care, and protected with careful engineering. There's a reason that flagship brands like Twilio, Peloton, Verizon, Nike, Dannon, MailChimp and others trust Netlify to host, scale, and deploy their mission critical sites.

In addition to features baked into the Netlify platform, Netlify Enterprise plans have additional security benefits, from penetration testing to support plans. Netlify Support Engineers are on-call 24x7 in case of an outage. Status of all systems is publicly available at all times on the Netlify status page. With an available Support Plan, organizations can escalate Critical and High Severity issues to Netlify Support Engineers by email or phone and receive a response in 60 minutes or less.

Contact Sales


netlify