netlify

Essential Security and Identity

Security and Identity for the Enterprise Using Netlify

Building with Confidence Using Netlify's Enterprise-Grade Security Tools

Trust is at the core of your business. It's hard earned, and carefully protected.

Netlify takes its responsibility as a platform that thousands of businesses, and hundreds of leading enterprises use to build, host, and deploy websites incredibly seriously.

In the digital age, your website acts as a proverbial front door to your business. It's the place your users know to come home to when they're using your service. It's a place they feel comfortable sharing sensitive information. With this trust users place in your site comes a duty to protect it. Netlify recognizes its critical role in that effort and has built enterprise-grade security practices and protocols to ensure that building on Netlify is not only secure but seamless.

Security was built into every decision we made in architecting the Netlify platform, and is present in every decision we make to further its capabilities. This security by design mentality means that safeguards are built into the platform as opposed to simply adding them onto the platform.

Netlify is SOC2 Type 2 certified, GDPR compliant, and regularly administers penetration tests performed by certified third parties, and employs security researchers to assess vulnerabilities on an ongoing basis. You can call this a best practice for a modern tech company using modern architecture. But, it's more than that to us. It's a commitment to our users, and our user's customers that we will protect the information and processes they entrust us with.

So, what exactly does this look like in action? Well, before we dive in, let's make sure you have a solid understanding of the security playing field starting with the most essential part of the Netlify platform — Jamstack.

Getting to Know Jamstack

It's easier to live up to modern security standards when you're using modern web architecture.

Netlify's Jamstack-based architecture decouples the frontend from the backend, giving you granular control of your website, more flexibility over what developer tools you use, and ensuring your sites perform at blazing speeds while running smoothly and securely.

In monolithic website architecture, the user's wait starts as soon as they hit enter in their browser's address bar. From there, they wait for their content to be ferried from their browser, across a CDN, through a load balancer, through various web servers, through a backend database and back again. This round trip from browser and back again typically relies on complex server side logic executions and the hope that all of the various vendors stitched together to comprise this framework are working well together. Needless to say, it's not as streamlined as it could be.

Using Jamstack, Netlify centralizes and streamlines the process of serving up a website or application using Javascript, APIs, and Markup. And, no Jamstack operation would be complete without static sites. While the Jamstack offers flexibility, the static site architecture therein offers users incredible speed.

In the Jamstack architecture, sites are prebuilt, waiting in the browser to be accessed by your users. So, the instant a user requests a site, it's there to greet them. This architecture gives you control and endless opportunity for extensibility with APIs, all without sacrificing speed or security.

Whether you're pushing content from a Headless CMS, or firing off a git push command from the command line, Netlify responds in kind. Netlify listens for changes to both content and code and responds accordingly when there's a change, executing build commands and updates based on rules you've dedicated.

Developers can integrate Netlify with top-tier git repo providers such as Github, Gitlab, or BitBucket to push code from their repo into the Netlify platform to deploy a new site or update an existing one.

Using Netlify, you can deploy to specific branches with a simple POST request to a webhook, or trigger notifications to specific Slack channels based on specific events in your build such as 'onSuccess'. The programmable possibilities are endless with Netlify.

The Netlify CDN

When you push sites live with Netlify, they're deployed on a globally available, reliable CDN with nodes around the world and secured end to end.

The CDN isn't just secure, it's smart, too. You can dictate the behavior of your traffic distributed across the CDN. For example, you could instruct your CDN to deliver a certain type of content to a user based on where they're based, what browser they're using, and if they've been separated into a split test you might have running. Netlify might rely on static sites, but there are a tremendous amount of dynamic functions you can run using Netlify.

Netlify's platform teams continuously monitor traffic patterns across Netlify's CDN to look out for fluctuations in both traffic and API request loads. By remaining vigilant and looking for any aberrations in network activity, Netlify can carefully spot and respond to threats in a matter of moments.

In this way, Netlify acts like another member of any customer's security team, helping to snuff out any issues before they state. In the event of something like a DDOS attack, Netlify has specific protocols in place to handle that event, carefully separating true web traffic from malicious traffic. By splitting well-meaning customers' traffic from bad actors' traffic, Netlify can more efficiently log and block the actors' IP addresses, shutting down the wouldbe attack.

The Netlify Build Environment

Malicious actors typically look for a window of opportunity to attack, be it a small window within a company's digital surface area, or a small window of time within a critical process.

The build process is often targeted by bad actors, so Netlify ensures any window of possible attack is completely shut, keeping every aspect of your website deployment secure. As a customer is building a website prior to its deployment on Netlify's CDN, Netlify spins up an ephemeral development environment that only exists for the moments in which the build is occurring. This way, there's no opportunity for a window of attack because the window is gone before any bad actors could spot it.

As most developers employ Netlify's API to run various build commands, Netlify allows you to dictate role-based permissions for API calls, so only people with permission to run something like a build command can do so. In addition to this role-based control, Netlify also employs volume limits on API-commands to spot and stop any nefarious activity before it can start.

Controlling Log In and Role-Based Permissions

Netlify gives you granular control over who can access your site, what parts of your site they can access, and what permissions they have. And, you can use the tools you know and love to dictate role-based access.

Netlify integrates easily with SAML 2.0 providers such as GSuite, Okta, OneLogin and other leading SAML providers, as well as various Two Factor Authentication and Multi-Factor Authentication providers like Authy.

With a network of identity verification software to use in conjunction with Netlify, you control how you authenticate users. Whether you want to vet users based on their git credentials, require them to login via SAML, dictate that their login must go through your SSO provider, or allow them to login via email, Netlify supports any and all of these workflows. Teams have to have the flexibility to work in a way that suits them, while not giving up an inch of security.

Accessing Logs Using Netlify

Imagine every employee at your company had a key to every room in your office. Nothing was off limits. Everything from the building's circuit breaker panel, to the locked office cabinets, to the private meeting rooms were all available for anyone to unlock. That wouldn't make much sense, right? The same principle applies when it comes to digital security.

So, revisiting our metaphor, it would make sense for your company's building supervisor to have a key to the circuit breaker panel, but it'd make less sense if the Head of PR was carrying it around. When we apply that principle in role-based access, we see how important it is for companies to be able to grant access to specific permissions, and areas of their infrastructure to specific employees based on what their job entails.

Netlify divides roles into three primary categories: Owners, Collaborators, and Billing Administrators.

Owners can do everything from creating and editing sites to modifying billing info, removing members, editing team settings, deleting teams, or transferring sites.

Collaborators can only create, view, edit, or change site addons.

Billing Admins are limited to modifying billing info and changing their team's Netlify subscription plan. When a user does take an action like editing billing settings, or creating a new site, Netlify keeps a log of those events so you always have a record to refer to.

Setting Site-Specific Permissions

When you're hard at work on a new product that's kept under wraps until it's ready to launch, you want to make sure that only your team of trusted colleagues can access that soon-to-be product's site. Netlify gives users intuitive and powerful tools to gate content by permissions both internally and externally.

In the example of a team working on a new product, you can choose to lock down a singular branch of your site, or a network of branches that relate to that product with Netlify's access controls. Now, when a colleague who isn't part of the product launch team attempts to access that site, they'll be met with a prompt to enter a password you've set. Without the password, they'll have to wait in suspense until the product launches. Another way to dictate access controls is by setting specific permissions in a site's Header or TOML file.

Netlify's access controls are so powerful and extensible that you could build an entire web application with nothing but Netlify, Stripe, Fauna and a series of cascading permission tiers. Let's say we wanted to create a site highlighting your favorite type of APIs.

A member who opts for the free tier of your API subscription site, might only be able to access three of the 20 APIs you love and recommend. But, how do you dictate that logic? With Netlify, you can define internal and external users' permissions to access sites using JSON WebTokens. So, if a premium user of your API highlight subscription site comes to your site and wants to browse the full list of APIs you recommend, you could access that user's ID and Stripe customer ID from your Fauna database, determine they're a premium member and let them see the full list of APIs.

Protecting Credentials and Variables

Maybe you'd keep your house keys under your doormat, or possibly a fake rock in a garden. When it comes to API keys and authentication tokens, you need much more stalwart security practices.

Using Netlify you can store sensitive information and variables that are best kept out of your repos. Storing those credentials so you can access them in ongoing CI / CD workflows is seamless. With a few simple commands, you can allow Netlify and your application to access those credentials automatically when executing various application commands. Now, you can reference those credentials in functions, in staging or branch deploys or other specific environments that you can choose. Netlify gives you complete control over the information that's tremendously valuable to your applications and protects it accordingly.

Netlify also supports JSON Web Signature to sign requests to your API or web service so you can be confident that requests to an external url such as your site originated from Netlify.

Build on a Secure, Scalable Platform

Netlify knows that trust is built with care, and protected with careful engineering. There's a reason that flagship brands like Peloton, Verizon, Nike, Dannon, MailChimp, Impossible Foods and others trust Netlify to host, scale, and deploy their mission critical sites. Security isn't just an area of focus for our company, it's built into every aspect of our company.

Whether you're deploying your first site on Netlify, or getting ready to deploy a few thousand sites, we are honored for the opportunity to support and secure your work.

Contact Us →

netlify