How Tremendous protects its software supply chain with Docker and Socket

Combining remediation at the base image with mitigation across open-source dependencies to secure the full software supply chain.

TREMENDOUS

Company: Tremendous

Industry: Financial technology (B2B rewards and payouts)

Headquarters: New York, New York

Scale: Over 20,000 organizations use Tremendous to reward users across 230 countries

Key Technologies: Docker Hardened Images, Socket (SCA and Firewall), Docker Desktop, Docker MCP Gateway, Ruby, Node.js, Google Cloud Artifact Registry

Challenge: Secure the full software supply chain, base images and open-source dependencies across its product

CHALLENGES

Securing the supply chain behind payouts for over 20,000 organizations

When a company sends a reward through Tremendous, it is trusting that the money arrives and that nothing rides along with it. From research teams at MIT and Google to marketing and HR programs at global brands, over 24,000 organizations use Tremendous to send payouts as money, prepaid cards, and gift cards to recipients across 230 countries. Tremendous has handled over 80 million payouts and more than $2 billion have moved across the platform, on a promise the company keeps easy: rewards and payouts, simplified.

As incidents across the open-source ecosystem continued to surge, the XZ Utils backdoor showed how a single trusted dependency could compromise everything built on top of it. Tremendous had already decided to tighten its software supply-chain controls, and customers were increasingly asking for stronger assurances around how production images were built, signed, and verified. Around the same time, the disclosure of the Trivy compromise reinforced that supply-chain risk could surface even through trusted security tooling.

The platform team, responsible for both infrastructure and security, had a clear goal: protect the full software supply chain, from the operating system in each base image to every open-source dependency layered above it.

For Tremendous, protecting the full attack surface meant securing two layers: the base image and the application dependencies built on top of it. Most tools covered only one.

A vulnerability could come from the operating-system packages shipped in the base image. It could also enter through a language dependency, such as a Ruby gem or npm package pulled in at install time.

> Docker's CVE justifications let us **present results to our engineering team credibly.** When something is marked non-applicable, the reasoning is right there, so we can focus remediation where it actually matters."
>
> **Lucas Alves**, Sr. Platform Engineer at Tremendous

Tremendous needed a solution that could secure both layers together, without forcing the team to stitch together separate tools or accept gaps in coverage. The right approach had to harden the container base, account for the open-source dependencies developers actually use, and give customers clear evidence that production images were protected and verifiable. That became the team's path forward: not just reducing vulnerabilities in one part of the stack, but building a more complete, auditable foundation for every image Tremendous ships.

SOLUTION

Remediation at the base image, mitigation across dependencies

Tremendous paired Docker Hardened Images, which remediate the base image, with Socket, which mitigates risk in the open-source dependencies on top of it. Docker Hardened Images provide a hardened, low-CVE base that a team cannot easily build and maintain on its own, shipped signed. Socket scans open-source dependencies in CI and at install time and blocks malicious or vulnerable packages before they enter a build, so the team can stop a threat immediately and decide on a patch on its own schedule. The two cover different layers, and they are built to work together: Docker Hardened Images ship with Socket Firewall built in.

> Pairing Docker's hardened images with Socket lets us **protect the base image and the open-source code on top of it as one supply chain.** We secure the foundation and the dependencies together, instead of hardening one layer and leaving the other exposed."
>
> **Lucas Alves**, Sr. Platform Engineer at Tremendous

Docker's handling of CVEs was the factor that decided the evaluation. When a vulnerability is marked non-applicable, Docker attaches the reasoning to the finding. That gives the platform team the context to take results back to engineers and act on them.

Here is how the two run in Tremendous's environment:

- **Hardened base images** replace the previous bases, shipping signed with a reduced CVE footprint.
- **Image customization** bakes the packages and certificates each service needs into the hardened image. Tremendous configured this through the UI first and is moving it to code-as-config.
- **Images are pulled through Tremendous's Google Cloud Artifact Registry** mirror, reconfigured to point at Docker's registry, so existing pull paths keep working with no change for developers.
- **Socket SCA** scans dependencies across repositories and CI, covering Ruby gems and npm packages.
- **Socket Firewall** blocks malicious or risky packages at install time, so a compromised dependency never reaches a build.

The Docker agreement also extended governance to an emerging need. Teams across Tremendous, including data, product, support, and marketing, build their own automations on internal systems that rely on LLMs. As these workflows become a regular part of how teams operate, Docker can give Tremendous the controls, usability, and visibility needed to support them safely. **Docker MCP Gateway can give the platform team a governed way to manage the local MCP tools these automations depend on,** while Docker Desktop will give teams visibility into what is running locally, moving them beyond a terminal-only setup that is difficult for non-technical users to inspect or manage.

MEASURABLE IMPACT

A security foundation that covers both layers of the supply chain

CVE reduction of at least 90%

Moving to hardened base images cut the known-vulnerability count on Tremendous's containers. Here's a breakdown of reductions:

Ruby core	Ruby test	Node core build	Node test build
95%	**90%**	**97%**	**95%**
CVE reduction	CVE reduction	CVE reduction	CVE reduction

Both layers of the supply chain covered together

Docker remediates the base image and Socket mitigates risk in the open-source dependencies, including Ruby and Node, so the foundation and the code on top of it are secured as one supply chain rather than by separate, unconnected tools.

Reduced image footprint and attack surface

Hardened images carry fewer packages, which reduces both image size and attack surface. Here's the breakdown:

Ruby runtime/core	Ruby build/test dependency layer	Node build stage
120	**170-200**	**80-100**
Fewer packages	Fewer packages	Fewer packages

Actionable CVE findings for the engineering team

Non-applicable vulnerabilities are flagged with Docker's reasoning attached to them, so the platform team can present scan results to engineers and direct remediation where it matters.

Proof of concept running in the first week

Tremendous had both Docker Hardened Images and Socket running in staging-equivalent repositories within the first week of evaluation. The migration to a new base image is intrinsic work, but the setup itself was straightforward.

RESULTS

A foundation that secures the whole supply chain

Tremendous set out to secure the full supply chain behind its payouts platform. With Docker remediating the base image and Socket mitigating risk across open-source dependencies, both layers are now covered. The rollout continues alongside the team's other work, moving image customization to code-as-config and extending hardened images across more of the platform.

The two products are built to work together. Docker Hardened Images ship with Socket Firewall, and the roadmaps are aligned, so coverage spans the whole surface without the integration gaps that come from stitching unrelated tools together. Docker's roadmap toward language-level dependency builds for ecosystems like Ruby and Node lines up with where Tremendous is headed next.

> Docker and Socket are built to work together, **which gives us confidence that the base image and the dependencies on top of it are covered as one supply chain.** As the threat landscape changes, we have a security foundation that can change with it."
>
> **Lucas Alves**, Sr. Platform Engineer at Tremendous