The Daily Swig
Cybersecurity news and views

Lessons learned: How a severe vulnerability in the OWASP ModSecurity Core Rule Set sparked much-needed change

Jessica Haworth 05 November 2021 at 15:47 UTC
Updated: 05 December 2022 at 16:01 UTC

Vulnerabilities WAF DevOps

Years-old WAF bypass flaw was discovered in June

A severe vulnerability present in the OWASP ModSecurity Core Rule Set (CRS) for several years was a “bang on the ear” for the project's maintainers, who have outlined steps to improve its security.

The ModSec team reported that a complete rule set bypass (CVE-2021-35368) had been discovered in June 2021. The critical bug that had been present in the software for at least four years.

As previously reported by *The Daily Swig*, the vulnerability bypassed the security protections offered by the in-built CRS web application firewall (WAF), meaning that malicious request body payloads could be smuggled through without being inspected.

The issue, which was introduced in code changes 2017, has since been remediated.

However, to help reduce the likelihood of another high-impact bug slipping through the net, the CRS maintainers have implemented new practices, guidelines, and a bug bounty program to further secure the technology.

A chance for growth

Speaking to *The Daily Swig*, OWASP CRS co-lead Christian Folini said that the incident was “one of the biggest bangs on the ear” for him.

He explained: “It was clearly my personal fault. I had introduced these two bugs right after our new team had taken over the dormant project in 2016. The fix was easy enough, yet I felt like hiding under a rock.”

BACKGROUND WAF bypass: ‘Severe’ OWASP ModSecurity Core Rule Set bug was present for several years

The team met in Switzerland last week to formulate a plan to “improve set up and procedures”, said Folini, who admitted that the incident was an “embarrassment”.

Folini said: “We started to look at it as a chance for growth and development. Therefore, we made it a focus theme for our first CRS developer retreat – how can we prevent merging bugs? And, if we carry them, how do we detect them in our rules?”

Beyond automated testing

Folini said that the CRS team has been slowly expanding its DevOps practices “for several years” since they took over in 2016.

However, the project is in need of “a comprehensive application security program that goes beyond automatic testing”, according to Folini.

Addressing the issue, he told *The Daily Swig* that the CRS team has implemented a list of changes that will foster a more proactive approach to security.

> “We ran a workshop at our retreat and tried to come up with various measures that would form several layers of defense against any failures in our rule set. Eventually we settled on four changes:
>
> - Rule exclusion packages will become plugins
>
> - We will forbid certain constructs/directives from the rule set
>
> - We will prepare a formal checklist to review every rule, namely new ones
>
> - We will launch a bug bounty program.”

Latest Posts

Deserialized web security roundup
Fortinet, Citrix bugs; another Uber breach; hacking NFTs at Black Hat

Critical IP spoofing bug patched in Cacti
‘Not that hard to execute if attacker has access to a monitoring platform running Cacti’

Casting a SpEL
Akamai WAF bypassed via Spring Boot to trigger RCE

Shrinking the attack surface

Folini explained that the bypass vulnerability was hidden in one of the rule exclusion packages, which are distributed together with the rule set.

“Even an inactive rule exclusion package could cripple the entire rule set,” he said.

“So, we want to transform these packages into plugins, leveraging the new plugin functionality that we are going to introduce with our next major release. Plugins are not installed by default, so this measure will reduce our attack surface.”

Read more of the latest security vulnerability news

Folini told *The Daily Swig* that the bypass was only possible because a bad rule used a “very powerful” construct to disable request body access under certain conditions.

“What we did not realize was that an attacker could meet these conditions by abusing the `PATH_INFO` part of the request URI,” he continued.

“If we forbid this and similarly powerful directives, we reduce the impact of future bugs, so this is a risk limiting measure.”

Folini also said that by introducing a formal checklist and a bug bounty program, code can be extensively reviewed, both internally and externally.

This plan does, however, contain a caveat: In order to be able to pay such bounties, the project will need to attract more sponsors.

Lessons learned

Despite the major security headache caused by the WAF bypass vulnerability, Folini is optimistic that the lessons learned will spark much-needed change.

“Nobody likes to look like an idiot in public,” he admitted. “But it is important to take the bull by the horns.

“Going out, writing a detailed advisory, informing the big integrators directly (in advance), responding to questions from the community and reviewing your own policies and procedures are all crucial steps if you want to grow and improve.

“The whole affair also strengthened the ties within the team. We overcame this together and we know we can trust each other even when we’re in deep trouble and that’s very reassuring.

“So ultimately, I think it was a positive experience and we trust that the community appreciates how we handled this severe incident. So, it looks like we’re doing the right thing here.”

RECOMMENDED ‘Focus on brilliance at the basics’ – GitHub CSO Mike Hanley on shifting left and securing the software supply chain

Vulnerabilities WAF DevOps DevSecOps Hacking culture Industry News Secure Development Interviews Bug Bounty Europe Open Source Software Switzerland Network Security Database Security Cloud Security

Jessica Haworth

@JesscaHaworth

Related stories

Deserialized web security roundup

Fortinet, Citrix bugs; another Uber breach; hacking NFTs at Black Hat

16 December 2022

Critical IP spoofing bug patched in Cacti

15 December 2022

Casting a SpEL

Akamai WAF bypassed via Spring Boot to trigger RCE

14 December 2022

Cloud flaws brought to the fore as bug bounty vulnerabilities hit 65k in 2022

13 December 2022

Burp Suite

Web vulnerability scanner
Burp Suite Editions
Release Notes

Vulnerabilities

Cross-site scripting (XSS)
SQL injection
Cross-site request forgery
XML external entity injection
Directory traversal
Server-side request forgery

Customers

Organizations
Testers
Developers

Company

About
PortSwigger News
Careers
Contact
Legal
Privacy Notice

Insights

Web Security Academy
Blog
Research
The Daily Swig

PortSwigger

Follow us