HUSKEYS

HUSKEYS RESEARCH · CVE-2026-114938

The Java4Shell Virtual Patching Playbook

A cross-vendor guide to patching, stop-gapping and virtually patching Java4Shell across Cloudflare, Fastly, Akamai, Imperva, AWS WAF, Azure, Google Cloud Armor and ModSecurity.

Roy Weisfeld
CTO, Huskeys

November 18, 2026
Status as of 08:00 UTC

huskeys.io
Network Edge Security Management

HUSKEYS

CONTENTS

What's inside

Java4Shell lets an unauthenticated attacker overflow the heap of a recent Java server with one malformed Authorization header, before any credential is checked. This playbook takes your team from triage to retirement of the virtual patch.

01 Java4Shell vulnerability overview — Page 3

02 JVM mitigation checklist — Page 4

03 Cross-vendor rule deployment — Page 5

04 Vendor-specific implementation instructions — Page 7

05 Validation and monitoring checklist — Page 12

06 When and how to retire the virtual patch — Page 13

Status as of November 18, 08:00 UTC

Item	Status
Oracle	Fixed in 17.0.21.1, 21.0.13.1, 25.0.5.1, 27.0.1.1
Amazon Corretto	Fixed in 17.0.21.1.2, 21.0.13.1.2, 25.0.5.1.2, 27.0.1.1.4
Azul Zulu	Fixed in 17.0.21.1, 21.0.13.1, 25.0.5.1
Eclipse Temurin	No fixed build yet
Public PoC	Crash and trigger proof-of-concept code on GitHub; Calif's RCE chains are still private
CISA KEV / in the wild	Not listed; no exploitation reported. Calif reports RCE on named products; Huskeys reproduced the heap overflow, not RCE.

This is moving quickly. Check your vendor's advisory before acting.

01 · OVERVIEW

Java4Shell vulnerability overview

A heap out-of-bounds write in OpenJDK's AVX-512 VBMI Base64 decoder, reachable before authentication through the Authorization header.

9.2 / 8.1
CVSS 4.0 (Calif) / CVSS 3.1 (Oracle)

62–63 bytes
written past the output array

Pre-auth
decoded before credentials or signature

How it works

To decode Base64 faster, OpenJDK uses a hand-tuned assembly intrinsic on CPUs with AVX-512 VBMI. When it meets `=` or `==` at the wrong length, it computes an output size of -1 and writes 62 or 63 fixed bytes past its output array. The published trigger is 64 `A` characters plus `=` on current JDK 21 and 25 builds, or 128 on older JDK 17 paths. Java then throws `IllegalArgumentException` and the app returns an ordinary 401, so a vulnerable and a patched server look identical from the outside.

Where it is reachable

Path	Exposed	Why
Basic auth: Spring Security, Tomcat, Jetty 12	Yes	`Base64.getDecoder()` runs before credentials are checked
Bearer JWT: Auth0 `java-jwt`	Yes	Header and payload decoded before signature verification
jjwt, nimbus-jose-jwt, Spring Security JWT	No	Own Base64 decoders
SAML, webhooks, Base64 body fields	Only with `java.util.Base64`	Not covered by edge rules on Authorization

Three gates

Hardware: x86-64 with AVX-512 VBMI and BW (Intel Ice Lake, AMD Zen 4 and newer). **Versions:** recent JDK 17, 21, 25 and 27; Java 8 and 11 are not affected per OpenJDK. **Warm-up:** the intrinsic is used after roughly 10,000 similar decodes.

02 · PATCH

JVM mitigation checklist

Patching the JVM is the only complete fix. Work through this list per host, starting with the ones that have the vulnerable instruction set.

Scope

- [ ] List every internet-facing Java service and its JDK vendor and version.
- [ ] Run the CPU check below on each x86-64 host; a result above 0 means in scope (AVX-512 BW is also required).
- [ ] Flag services that decode Authorization with Spring Security Basic auth, Tomcat or Jetty 12 `BasicAuthenticator`, or Auth0 `java-jwt`.

```bash
grep -c avx512vbmi /proc/cpuinfo
```

Patch or stop-gap

- [ ] Upgrade to Oracle 17.0.21.1 / 21.0.13.1 / 25.0.5.1 / 27.0.1.1, Corretto 17.0.21.1.2 / 21.0.13.1.2 / 25.0.5.1.2 / 27.0.1.1.4, or Zulu 17.0.21.1 / 21.0.13.1 / 25.0.5.1.
- [ ] Temurin or other builds without a fix: start the JVM with `-XX:-UseBASE64Intrinsics`.
- [ ] Alternative on x86-64 only: `-XX:UseAVX=2` (caps vector instructions for the whole JVM).

Do not use `-XX:DisableIntrinsic=...` on its own: without `-XX:+UnlockDiagnosticVMOptions` the JVM refuses to start. `-XX:UseAVX=2` is x86-64 only; an ARM64 JVM will not start with it, and ARM hosts need neither flag.

Confirm

- [ ] `java -version` shows a fixed build, or the flag is present on the running process.
- [ ] Record the host, build and date in your change log, so the edge rule can be retired later.

03 · DEPLOY

Cross-vendor rule deployment

Block Authorization tokens that are already invalid Base64. Valid padded Base64 is always a multiple of 4 and real JWTs are unpadded, so false positives are limited to clients that already send broken tokens.

1
Normalize
Lowercase the Authorization value

2
Shape
≥32 chars, ends in `=` or `==`

3
Length
Not divisible by 4

4
Act
Block Basic and Bearer; log the rest

J4S-BASIC · Block

REGEX

```
^[ \t]*basic.[ \t]*(?:[a-z0-9+/]{4}){7,}(?:[a-z0-9+/]{3}==|[a-z0-9+/]{4}(?:[a-z0-9+/]{0,2}=|[a-z0-9+/]?==))(?:[^= \t]|[ \t]*$)
```

J4S-BEARER · Block

REGEX

```
^[ \t]*bearer.[ \t]*(?:[^. \t]*[.]){0,2}(?:[a-z0-9_-]{4}){7,}(?:[a-z0-9_-]{3}==|[a-z0-9_-]{4}(?:[a-z0-9_-]{0,2}=|[a-z0-9_-]?==))(?:[.]|[ \t]*$)
```

J4S-ANY · Log / Count only

REGEX

```
^[ \t]*[a-z][a-z0-9_-]*.[ \t]*(?:[a-z0-9+/_-]{4}){7,}(?:[a-z0-9+/_-]{3}==|[a-z0-9+/_-]{4}(?:[a-z0-9+/_-]{0,2}=|[a-z0-9+/_-]?==))(?:[^= \t]|[ \t]*$)
```

Tested. Matched 285 of 285 overflowing inputs and 0 of 24,000 legitimate tokens. Identical results in Python `re`, google-re2 and PCRE2 on a 50,295-line corpus. Line wraps in this PDF are visual only: paste each pattern as a single line.

03 · DEPLOY

Coverage at a glance

No major edge vendor had shipped a managed Java4Shell rule as of November 18, 08:00 UTC. Deploy the custom rules now and switch to the vendor rule when it ships.

Cloudflare

None found

CUSTOM RULE
Regex custom rule, Block

LIMITATION
Free and Pro plans have no `matches` operator

Fastly

None found

CUSTOM RULE
Regexp request rule, Block

LIMITATION
Header match is case-sensitive; keep `(?i)`

Akamai

None found*

CUSTOM RULE
Property Manager regex, 403

LIMITATION
App & API Protector rules cannot express the length check

Imperva

None found

CUSTOM RULE
Partial Bearer rule

LIMITATION
No header regex; misses Basic and the published trigger

AWS WAF

None found

CUSTOM RULE
Regex pattern sets

LIMITATION
About 35 WCU per rule

Microsoft Azure WAF

None found

CUSTOM RULE
RegEx MatchRule, Block

LIMITATION
CRS 3.1 and older reject spaces in values

Google Cloud Cloud Armor

None found

CUSTOM RULE
CEL rule, preview first

LIMITATION
Repeated headers are joined with commas

ModSecurity / OWASP CRS

No CRS rule

CUSTOM RULE
Local SecRules, phase 1

LIMITATION
Local rule IDs to maintain

* Akamai check is bounded: the Akamai blog blocks our fetchers, and web search found nothing.

Rollout order

1. Map every edge in front of a Java origin: CDN, cloud WAF, load balancer, ingress.
2. Deploy J4S-BASIC and J4S-BEARER in Block and J4S-ANY in Log on every path to the origin.
3. When a vendor ships a managed rule, enable it and keep ours in Log for a day to compare.

04 · VENDORS

Cloudflare

CLOUDFLARE

MANAGED PROTECTION

None found. The WAF changelog's latest entry is still November 4; the changelog RSS has no WAF items for November 16–18.

CUSTOM RULE

On Business or Enterprise, create a WAF custom rule with action Block. On Enterprise, add a second rule with action Log for J4S-ANY.

CLOUDFLARE · BLOCK

```
any(lower(http.request.headers["authorization"][*])[*] matches r"^[ \t]*basic.[ \t]*(?:[a-z0-9+/]{4}){7,}(?:[a-z0-9+/]{3}==|[a-z0-9+/]{4}(?:[a-z0-9+/]{0,2}=|[a-z0-9+/]?==))(?:[^= \t]|[ \t]*$)")
or
any(lower(http.request.headers["authorization"][*])[*] matches r"^[ \t]*bearer.[ \t]*(?:[^. \t]*[.]){0,2}(?:[a-z0-9_-]{4}){7,}(?:[a-z0-9_-]{3}==|[a-z0-9_-]{4}(?:[a-z0-9_-]{0,2}=|[a-z0-9_-]?==))(?:[.]|[ \t]*$)")
```

CLOUDFLARE · LOG

```
any(lower(http.request.headers["authorization"][*])[*] matches r"^[ \t]*[a-z][a-z0-9_-]*.[ \t]*(?:[a-z0-9+/_-]{4}){7,}(?:[a-z0-9+/_-]{3}==|[a-z0-9+/_-]{4}(?:[a-z0-9+/_-]{0,2}=|[a-z0-9+/_-]?==))(?:[^= \t]|[ \t]*$)")
```

LIMITATIONS

Free and Pro plans have no `matches` operator. Rely on the origin JVM flag there.

04 · VENDORS

Fastly and Akamai

fastly

MANAGED PROTECTION

None found; the Fastly blog has no Java4Shell post.

CUSTOM RULE

Next-Gen WAF: add one request rule per pattern on header `authorization`, operator Matches (regexp), action Block.

REGEX · ONE RULE PER LINE

```
(?i)^[ \t]*basic.[ \t]*(?:[a-z0-9+/]{4}){7,}(?:[a-z0-9+/]{3}==|[a-z0-9+/]{4}(?:[a-z0-9+/]{0,2}=|[a-z0
-9+/]?==))(?:[^= \t]|[ \t]*$)
(?i)^[ \t]*bearer.[ \t]*(?:[^. \t]*[.]){0,2}(?:[a-z0-9_-]{4}){7,}(?:[a-z0-9_-]{3}==|[a-z0-9_-]{4}(?:
[a-z0-9_-]{0,2}=|[a-z0-9_-]?==))(?:[.]|[ \t]*$)
```

LIMITATIONS

Header matching is case-sensitive, so keep the `(?i)` prefix. Verify the regex flavor in the console before production.

Akamai

MANAGED PROTECTION

None found. Bounded check: the Akamai blog blocks our fetchers, and web search found nothing.

CUSTOM RULE

1. Property Manager: Set Variable `PMUSER_J4S_AUTH` from request header `Authorization` (Lowercase, mark sensitive).
2. Add Regular expression criteria on `{{user.PMUSER_J4S_AUTH}}` with J4S-BASIC and J4S-BEARER.
3. Construct Response 403. Stage first, then activate.

LIMITATIONS

App & API Protector custom rules only support wildcards, which cannot express "length not a multiple of 4".

04 · VENDORS

Imperva and AWS WAF

Imperva

MANAGED PROTECTION

None found; Imperva blog posts through November 12 do not cover it.

CUSTOM RULE

Match a Bearer JWT whose header or payload segment is padded. Click Validate before saving, run in Alert for 24 hours, then switch to Block.

INCAPRULES

```
HeaderValue contains {"Authorization";"^bearer"} & HeaderValue contains {"Authorization";"=."}
```

LIMITATIONS

No regex on headers, so this covers one shape only. It misses Basic auth, end-of-token padding (the published trigger) and signature padding. Ask Imperva support for a managed rule.

AWS WAF

MANAGED PROTECTION

None found. The November 17 managed rule group entries are Bot Control 2.1 and POSIX OS 3.3.

CUSTOM RULE

1. Create regex pattern set `java4shell-auth-block` with J4S-BASIC and J4S-BEARER; Block rule on single header `authorization`, LOWERCASE transform.
2. Create `java4shell-auth-any-count` with J4S-ANY; Count rule, same match.

LIMITATIONS

About 35 WCU per rule (25 for the pattern set reference, 10 for LOWERCASE). Check web ACL capacity first.

04 · VENDORS

Azure WAF

Microsoft Azure WAF

MANAGED PROTECTION

No Java4Shell rule in DRS / CRS or MS-ThreatIntel-CVEs.

CUSTOM RULE

Front Door and Application Gateway: MatchRule on RequestHeader `Authorization`, operator RegEx, transform Lowercase, match values J4S-BASIC and J4S-BEARER, action Block. Validate `^` and `$` in the portal before rollout.

JSON · CRS 3.1 AND OLDER

```
^\\s*basic.\\s*(?:[a-z0-9+/]{4}){7,}(?:[a-z0-9+/]{3}==|[a-z0-9+/]{4}(?:[a-z0-9+/]{0,2}=|[a-z0-9+/]?==))(?:[^=\\s]|\\s*$)
^\\s*bearer.\\s*(?:[^.\\s]*[.]){0,2}(?:[a-z0-9_-]{4}){7,}(?:[a-z0-9_-]{3}==|[a-z0-9_-]{4}(?:[a-z0-9_-]{0,2}=|[a-z0-9_-]?==))(?:[.]|\\s*$)
```

LIMITATIONS

Application Gateway policies on CRS 3.1 or older reject spaces in match values; use the JSON-escaped values above.

04 · VENDORS

Google Cloud Armor and ModSecurity

Google Cloud Armor

MANAGED PROTECTION

None found; preconfigured WAF rules (java-v422) have no Java4Shell rule.

CUSTOM RULE

Create the rule with `--preview` on `gcloud compute security-policies rules create`, check the logs, then `rules update --no-preview`.

CEL

```
has(request.headers['authorization']) && (request.headers['authorization'].lower().matches(r'^[ \t]*basic.[ \t]*(?:[a-z0-9+/]{4}){7,}(?:[a-z0-9+/]{3}==|[a-z0-9+/]{4}(?:[a-z0-9+/]{0,2}=|[a-z0-9+/]?==))(?:[^= \t]|[ \t]*$)') || request.headers['authorization'].lower().matches(r'^[ \t]*bearer.[ \t]*(?:[^. \t]*[.]){0,2}(?:[a-z0-9_-]{4}){7,}(?:[a-z0-9_-]{3}==|[a-z0-9_-]{4}(?:[a-z0-9_-]{0,2}=|[a-z0-9_-]?==))(?:[.]|[ \t]*(?:,|$))'))
```

LIMITATIONS

Cloud Armor joins repeated headers into one comma-separated value, so the Bearer pattern also accepts a trailing comma.

ModSecurity / OWASP CRS

MANAGED PROTECTION

No CRS rule found.

CUSTOM RULE

Local SecRules in phase 1 with `t:lowercase`: deny for J4S-BASIC and J4S-BEARER, pass and log for J4S-ANY.

LIMITATIONS

Use IDs in the local-use range, and remove the rules once CRS ships coverage.

05 · VALIDATE

Validation and monitoring checklist

Status codes will not tell you whether an attempt landed: vulnerable and patched servers both answer 401. Validate the rule, then watch the hosts.

Before Block

- [ ] Send a synthetic trigger (64 `A` characters plus `=` as a Basic token) from a test client; confirm it is blocked on every edge.
- [ ] Send valid Basic credentials and a real JWT; confirm both pass.
- [ ] Confirm the rule appears in each platform's logs with the expected action.

First 24 hours

- [ ] Watch 401 and 403 rates per edge and per application.
- [ ] Review J4S-ANY matches for scanners probing other auth schemes.
- [ ] Investigate any client that legitimately sends padded, invalid tokens and add a narrow exception.

Hunting on the origin

- [ ] Search application logs for `Input byte array has wrong 4-byte ending unit`.
- [ ] Look for new `hs_err_pid*.log` files or unexpected JVM restarts since November 16.
- [ ] Run J4S-ANY over retained Authorization values, after confirming the log path keeps the header.

> **Log coverage.** Some log paths drop the Authorization header entirely. A clean hunt over a path that drops it proves nothing.

06 · RETIRE

When and how to retire the virtual patch

A virtual patch is a snapshot, not a control. Remove it once it is no longer needed, so it does not become a forgotten rule.

Retire when all of these are true

- [ ] Every in-scope host runs a fixed JDK build, confirmed with `java -version`.
- [ ] No host depends on a JVM flag that a redeploy could drop.
- [ ] New images and autoscaling templates use the fixed build.
- [ ] The vendor managed rule, where available, is enabled in Block.

How to retire

1. Move J4S-BASIC and J4S-BEARER from Block to Log on one edge first.
2. Watch for 24 hours: J4S-ANY and the vendor managed rule (if any) should show the same picture.
3. Remove the custom rules edge by edge, recording each change.
4. Keep the vendor managed rule in Block if one has shipped.
5. Close the change with the host, build and rule inventory attached.

> **Do not retire early** if any origin is still running an unpatched build with only a JVM flag that could be lost on redeploy or autoscaling.

HUSKEYS

CHECK YOUR EDGE

One detection, every edge.

Writing the rule is the easy part. Knowing where it needs to go, and keeping it consistent, is the hard part. Huskeys is a Network Edge Security Management (NESM) layer on top of the edge platforms you already run.

01 **Know your estate.** Which edges you run, which Java origins sit behind each one, and who owns every layer.

02 **Find the gaps.** Where managed coverage exists, where a custom rule is needed, and where only a partial one is possible.

03 **Deploy consistently.** One detection, translated and tested for each platform, rolled out in Log and then Block, checked against your real traffic.

04 **Keep it current.** Switch to vendor-managed rules as they ship, and retire the virtual patch once every origin is fixed.

See which of your edges would stop Java4Shell today.

Book a Demo