ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 40 / 30 · ENTERPRISE EXPLOITATION

JimuReport Probes, Polyglot JSPs, and WordPress Race Tooling

Three Exposed Workbenches

How three exposed APAC directories pair Java reporting scanners and image-header JSP shells with a WordPress core upgrade-race exploit

Prepared November 2026
Full Report · Public edition · Version 1.1

3
Open HTTP directories in China and Hong Kong modeled in this study (0 shared hosts)

1,039
Total exposed files reported across the three directories (988, 47, and 4)

269
Sampled directory entries reviewed (240 unique SHA-256 digests, 10 digests withheld at source)

14
Core RCE, upgrade-race, polyglot JSP webshell, and WordPress shell-deployment files

What Is in This Report. 11 chapters and two appendices.

How three exposed APAC directories pair Java reporting scanners and image-header JSP shells with a WordPress core upgrade-race exploit

01	Executive Summary	04
02	What Was Found: 269 Sampled Entries Across Three Open Directories	07
03	Workbench A: JimuReport and Quartz RCE Scripts With Polyglot JSP Webshells	11
04	Workbench B: Racing WordPress Core Upgrades Alongside Brute-Force Tooling	14
05	Workbench C: Call-Center Web-Phone Probes and Enumeration Logs	17
06	Analytical Model Boundaries and Alternative Explanations	20
07	What Defenders Should Do	22
08	Detection Rules	24
09	Indicators	27
10	Scope and Limits	29
11	Sources	31
A	Indicators of compromise (0)	33
B	Detection rules (4 files)	34

IN SHORT

From 13 to 21 November 2026, Odin by Brandefense reviewed 269 sampled files across three open web directories in China and Hong Kong reporting 1,039 files. The servers stage Python scanners for JeecgBoot JimuReport and Java Quartz schedulers, GIF and JPEG polyglot JSP webshells, and a race exploit for WordPress Core upgrades. Patch JimuReport to 1.6.1 or later, update WordPress to 7.1.4 or fixed branch releases, block external requests to upgrade folders, and re-encode uploaded images.

01

CHAPTER 01

Executive Summary

IN SHORT

From 13 to 21 November 2026, Odin by Brandefense reviewed 269 sampled files across three open web directories in China and Hong Kong reporting 1,039 files. The servers stage Python scanners for JeecgBoot JimuReport and Java Quartz schedulers, GIF and JPEG polyglot JSP webshells, and a race exploit for WordPress Core upgrades. Patch JimuReport to 1.6.1 or later, update WordPress to 7.1.4 or fixed branch releases, block external requests to upgrade folders, and re-encode uploaded images.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

Many enterprise networks run Java reporting services and scheduled job engines next to PHP content management systems behind the same web proxies. That setup speeds up business reporting and publishing, yet it widens the external attack surface across template routes, upload forms, and maintenance folders.

This report examines 269 sampled entries across three open HTTP directories in Asia-Pacific to show how attackers target both stacks. We did not run any recovered scripts, and we did not contact any server named in the files.

1. **Java reporting scanners sit beside image-header polyglot JSP webshells.**
   One open directory in China (port withheld) reports 988 files. Its 218 sampled entries include Python scripts written to test JeecgBoot JimuReport flaw CVE-2023-4450, which defenders can prioritize with vulnerability intelligence, and Java Quartz job schedulers. Next to Fastjson, Apache Shiro, and Nacos scripts sit two JSP webshells that start with GIF and JPEG image headers to slip past basic upload checks.

2. **A Hong Kong workbench stages a WordPress core upgrade-race exploit.**
   Observed on 20 November 2026 in Hong Kong (port withheld), a 47-file directory holds a 4,377-byte Python race script for CVE-2026-119977, published in NVD on 31 October 2026. WordPress Core 4.7 through 7.1.3 uses predictable, web-reachable staging paths during full-package core upgrades. Co-hosted scripts handle WordPress login brute-forcing, XML-RPC multicall amplification, backup checks, FTP tests, and webshell uploads.

3. **More than half of the sampled files are client bundles or stubs.**
   Out of 269 sampled entries and 240 unique SHA-256 digests, 152 files are unpacked mini-program components, duplicated webpack chunks, an open-source WordPress CLI archive, and a nine-byte HTTP 404 stub. Another 12 files are password wordlists and call-center portal captures from a third server (port withheld). Because directory ownership is unestablished, all file hashes stay withheld.

Three Decisions for Leaders

1. **Patch Java reporting engines and WordPress core across every tier.** Update JeecgBoot JimuReport to 1.6.1 or later, remove public access to Quartz scheduler routes, and upgrade WordPress to 7.1.4, 7.0.8, 6.9.11, or your branch's fixed release while blocking `/wp-content/upgrade/` at the perimeter.
2. **Stop trusting file extensions and leading image bytes on uploads.** Store user uploads outside Java servlet and PHP web roots, disable script handlers on media folders, and re-encode uploaded GIF and JPEG images on the server.
3. **Filter downloaded client bundles and HTTP stubs before feeding blocklists.** Keep unpacked front-end JavaScript libraries, open-source administration archives, wordlists, and reconnaissance logs out of automated EDR hash blocks.

The Study in Numbers

Number	What it is
3	Open HTTP directories in China and Hong Kong modeled in this study (0 shared hosts)
1,039	Total exposed files reported across the three directories (988, 47, and 4)
269	Sampled directory entries reviewed (240 unique SHA-256 digests, 10 digests withheld at source)
14	Core RCE, upgrade-race, polyglot JSP webshell, and WordPress shell-deployment files
91	Supporting Python, shell, and PHP scripts for login brute force, XML-RPC, SQLi, IDOR, and FTP
164	Non-indicator files kept out of hash feeds (152 bundles and stubs plus 12 wordlists and logs)
0	File hashes or host addresses published in Appendix A (withheld pending ownership confirmation)

How to use this report.

Security leaders can start with Chapters 1, 6, and 7. Java platform, CMS, and WebOps engineers should read Chapters 3, 4, and 7. SOC analysts and detection engineers should go to Chapters 8 and 9. Analysts using cyber threat intelligence will find the three-workbench breakdown and model limits in Chapters 2, 5, 6, and 10.

02

CHAPTER 02

What Was Found: 269 Sampled Entries Across Three Open Directories

IN SHORT

Three open HTTP directories in China and Hong Kong reported 1,039 exposed files, of which 269 entries were sampled. Static review separates 14 core exploit and webshell files and 91 supporting scripts from 164 front-end bundles, stubs, wordlists, and logs.

INSIDE THIS CHAPTER

Three distinct workbenches 08

Five classes of file 08

Why raw directory hashes make bad blocklists 09

02 What Was Found: 269 Sampled Entries Across Three Open Directories

Firewall logs show individual exploit requests arriving at a web server. Open HTTP directories left on virtual private servers show the fuller workspace behind those requests. They reveal how an operator pairs vulnerability scanners with upload-bypass webshells, wordlists, and downloaded target scripts.

Odin by Brandefense recorded three such directories between 13 and 21 November 2026. The three hosts share zero IP addresses, zero file hashes, and zero script names. Because no technical indicator links them to one operator, we present them as an analytical model of current web and middleware intrusion tooling in Asia-Pacific. All three server addresses and ports are withheld.

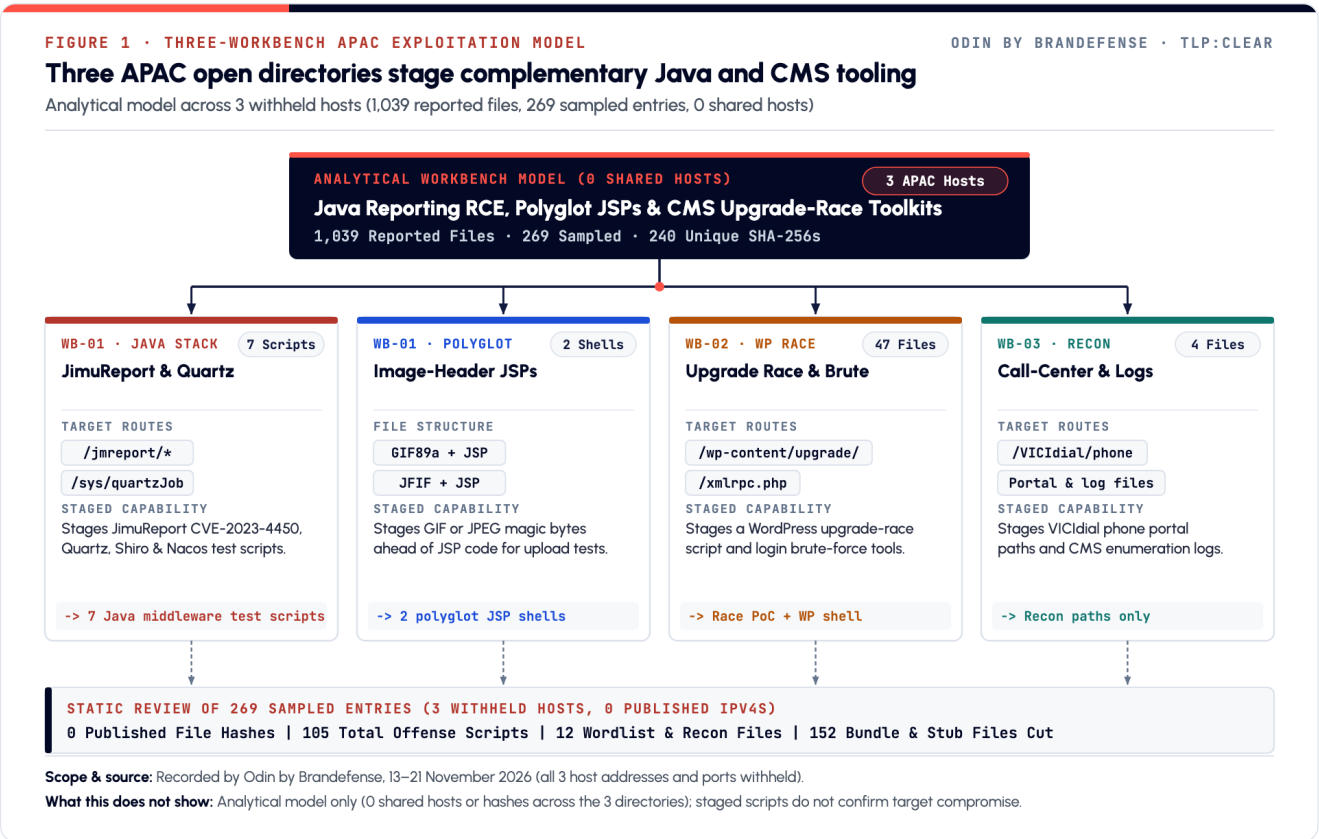


Figure 1: Three APAC open directories stage complementary Java and CMS tooling

Three distinct workbenches

Workbench A in China (port withheld) reports 988 exposed files, with 218 entries sampled between 13 and 20 November 2026. Workbench B in Hong Kong (port withheld) exposes 47 files. All 47 were indexed on 21 November 2026 after the directory was first seen late on 20 November. Workbench C in China (port withheld) exposes 4 text entries, all 4 sampled on 20 November 2026.

Workbench	Location	Reported files	Sampled entries	Unique SHA-256	Digests withheld at source	Main contents
Workbench A (`WB-01`)	China (port withheld)	988	218	191	8	JimuReport and Quartz RCE scripts, GIF/JPEG JSP shells, and unpacked client bundles
Workbench B (`WB-02`)	Hong Kong (port withheld)	47	47	46	1	WordPress CVE-2026-119977 race script, CMS exploit script, XML-RPC brute tools, and FTP helpers
Workbench C (`WB-03`)	China (port withheld)	4	4	3	1	`/VICIdial/phone` call-center path, portal login probe, backslash probe, and enumeration log

Five classes of file

Across all 269 sampled entries, there are 149 JavaScript files, 90 Python scripts, 9 text files, 7 shell scripts, and 4 PHP files. The remaining entries are 4 extensionless files, 2 JSP webshells, 1 HTML page, 1 JSON catalog, 1 PHAR archive, and 1 compiled `.pyc` file. Two hundred fifty-nine entries carry recorded SHA-256 digests, and 10 login or portal entries have no published digest.

Deduplicating the 259 hashed rows leaves 240 unique SHA-256 digests. All 19 duplicate pairs sit on Workbench A, where 19 compiled Vue webpack chunks in one front-end folder appear a second time byte-for-byte in a second front-end folder (paths withheld).

Artifact class	Sampled files	Share	Unique SHA-256	Hosts	Size range (bytes)	What it contains
RCE and race exploits	10	3.7%	10	2	1,092 to 84,986	Java reporting/middleware test scripts and WordPress upgrade-race tooling
Webshells and droppers	4	1.5%	4	2	1,374 to 2,172 (JSPs chunked)	GIF and JPEG polyglot JSP webshells plus two WordPress shell uploaders
Auth, brute, and pivot scripts	91	33.8%	84	2	294 to 5,658	Login brute-forcers, XML-RPC multicall, SQLi, IDOR, upload, and FTP helpers
Wordlists and recon captures	12	4.5%	11	2	379 to 829,702	Password mask lists plus VICIdial and CMS enumeration files
Bundles, packages, and stubs	152	56.5%	131	2	9 to 7,142,777	Unpacked mini-program/H5 JavaScript, WordPress CLI PHAR archive, and a 9-byte 404 stub

Two polyglot JSP rows on Workbench A (GIF-header and JPEG-header JSP files; filenames withheld) record a content length of -1. The HTTP server sent them without a fixed `Content-Length` header, while their bodies still hashed to two distinct SHA-256 digests.



Figure 2: Static review separates 14 core exploit files from 152 bundle files

Why raw directory hashes make bad blocklists

Scraping every hash from an open directory creates immediate false positives. One hundred fifty-two of the 269 sampled entries (56.5%) are ordinary front-end JavaScript bundles, open-source tools, or HTTP error text.

On Workbench B, a 7,142,777-byte PHAR archive is the legitimate open-source WordPress command-line administration tool. Right beside it sits a 9-byte HTTP error stub (path withheld). Although its directory path on the server borrows the name of a public PHP-FPM exploit tool, its 9-byte content is the literal ASCII string `Not Found`. Blocking that SHA-256 digest in an EDR tool would flag every 9-byte `Not Found` file across an enterprise.

CHAPTER 03

Workbench A: JimuReport and Quartz RCE Scripts With Polyglot JSP Webshells

IN SHORT

Workbench A targets Java enterprise stacks built around JeecgBoot. It combines Python scanners for JimuReport CVE-2023-4450 and Quartz job schedulers with GIF and JPEG polyglot JSP webshells designed to bypass simple image upload filters.

INSIDE THIS CHAPTER

Probing JimuReport (`CVE-2023-4450`) and Java Quartz schedulers 11

Supporting API, SQL injection, and authentication test scripts 12

How GIF and JPEG polyglot JSP webshells bypass naive upload filters 12

Unpacked mini-program and H5 front-end bundles 12

03 Workbench A: JimuReport and Quartz RCE Scripts With Polyglot JSP Webshells

Across Asia-Pacific, many internal portals and public business platforms use Spring Boot low-code stacks such as JeecgBoot. A typical deployment bundles JimuReport for data dashboards, Quartz for scheduled background jobs, and Fastjson for JSON parsing. It also pairs Apache Shiro for session cookies and Nacos for service configuration with a UniApp or WeChat mini-program client.

Workbench A (China, port withheld) lines up directly with that stack. Out of 218 sampled entries, 68 are Python, shell, bytecode, and JSP offense files, while 150 are downloaded front-end JavaScript and HTML files.

Probing JimuReport (`CVE-2023-4450`) and Java Quartz schedulers

Two Python scripts sit at the core of Workbench A's server-side tooling: a 3,199-byte JimuReport probe and a 4,464-byte Quartz probe (filenames withheld). Both carry file observation dates from 13 to 20 November 2026 (T1190).

The 3,199-byte script targets **CVE-2023-4450**, a critical template injection flaw in JeecgBoot JimuReport. NVD lists JimuReport 1.0 through 1.6.0 as affected by CVE-2023-4450. The flaw is in the Template Handler and was published on 21 August 2023. NVD rates it 9.8 Critical under CVSS v3.1; VulDB rates it 6.3 Medium.

Crafted HTTP POST requests to routes such as `/jmreport/queryFieldBySql` or `/jmreport/show` can pass FreeMarker template expressions (`freemarker.template.utility.Execute`). Those requests can also invoke Java runtime calls that run operating-system commands. JeecgBoot fixed the vulnerability in JimuReport **1.6.1**.

The 4,464-byte script targets Java Quartz job-scheduler interfaces, such as `/sys/quartzJob/` or REST `/scheduler/jobs` routes often exposed on Spring Boot admin panels. Weak scheduler endpoints let untrusted callers create or edit jobs without strict authentication or class allowlists. An attacker can then submit a `jobClassName` or script parameter that invokes `java.lang.Runtime`, `ProcessBuilder`, or JNDI `InitialContext` when the trigger fires.

Five more scripts on Workbench A test adjacent Java middleware weaknesses:

Script role (filename withheld)	Size (bytes)	Target component and role
JimuReport RCE test script (Python)	3,199	JeecgBoot JimuReport 1.0 to 1.6.0 Template Handler RCE (`CVE-2023-4450`)
Quartz job RCE test script (Python)	4,464	Java Quartz scheduler job-registration command execution
Fastjson gadget test script (Python)	1,369	Alibaba Fastjson deserialization gadget testing
Shiro cipher key scripts (2 Python files)	2,707 and 2,710	Apache Shiro `RememberMe` cipher key brute-forcing and verification
Nacos and middleware SSRF scripts (2 shell files)	1,092 and 1,535	Alibaba Nacos and middleware server-side request forgery checks

Supporting API, SQL injection, and authentication test scripts

Alongside the seven middleware test scripts, Workbench A holds 59 supporting Python, shell, and `.pyc` utilities (618 to 5,658 bytes; filenames withheld). Several target JeecgBoot conventions directly, including a 2,700-byte dictionary lookup tester and a 2,842-byte request-signature tester. The folder also holds login test scripts (including Python 3.13 bytecode) and two admin brute-force tools.

Other scripts test boolean and UNION SQL injection (5,658 bytes and 2,044 bytes). Companion utilities test JWT secret guessing (2,725 bytes) and OAuth grant handling (2,446 bytes).

Additional helpers test padding-oracle recovery, CAPTCHA reuse, insecure direct object references, and automated record collection (5,486 bytes). Because owners of the server and custom API test scripts are not established, we withhold all file paths and hashes.

How GIF and JPEG polyglot JSP webshells bypass naive upload filters

Workbench A also stages two polyglot JSP webshells next to upload-testing scripts (2,192 bytes and 2,159 bytes; filenames withheld). Additional upload helpers include a 3,545-byte extension-whitelist tester, a 1,708-byte HTTP `PUT` tester, and a 953-byte upload shell script.

A polyglot JSP webshell is a single file that is valid for two parsers at once (T1505.003). Its opening bytes start with a standard image signature: `GIF89a` (or `GIF87a`) for GIF, or `FF D8 FF` (`JFIF`) for JPEG. Upload validators that inspect only the `Content-Type` header or the first few magic bytes classify the file as a harmless image.

Later in the same file sits a `<% ... %>` JSP scriptlet calling Java execution APIs such as `Runtime.getRuntime()` or `ProcessBuilder`. An attacker pairs that file with an extension-whitelist flaw, a path-parameter trick, or an open HTTP `PUT` route. Once saved as a `.jsp` file inside a servlet-parsed folder, Tomcat or Jetty treats the image header as template text and compiles the JSP code.

Unpacked mini-program and H5 front-end bundles

The remaining 150 sampled entries on Workbench A are downloaded client-side files rather than server exploits:

- **Unpacked mini-program bundle (110 files).** Stored in an unpacked mini-program folder (path withheld), this tree holds 109 JavaScript files and 1 HTML login page (452 bytes). It includes Babel runtime helpers, 47 `wot-design-uni` user-interface components, and department user-picker widgets. It also holds an MD5 request-signing helper (filename withheld) and three voice/translation plugin files whose mini-program AppID is withheld.
- **Duplicated H5 webpack chunks (38 files) and root bundles (2 files).** Two shell helpers (1,017 and 1,244 bytes; filenames withheld) support front-end downloading and route probing. Beside them, two front-end folders hold 19 identical pairs of Vue webpack chunks (up to 814,735 bytes; paths withheld). Two more 300,001-byte JavaScript bundles sit at the directory root.

Attackers unpack mini-programs and webpack chunks offline to read hidden API paths and request-signing logic before running server probes. None of these 150 client files is published as an indicator hash.

CHAPTER 04

Workbench B: Racing WordPress Core Upgrades Alongside Brute-Force Tooling

IN SHORT

Observed on 20 November 2026 in Hong Kong, Workbench B stages tooling associated with WordPress Core CVE-2026-119977. Other files include CMS tests, XML-RPC brute-forcers, webshell uploaders, and FTP helpers.

INSIDE THIS CHAPTER

Racing predictable staging paths in CVE-2026-119977 14

Fallback CMS exploits, XML-RPC brute-forcing, and pivot helpers 15

04 Workbench B: Racing WordPress Core Upgrades Alongside Brute-Force Tooling

While Workbench A focuses on Java middleware, Workbench B (Hong Kong, port withheld) targets PHP and WordPress environments across 47 exposed files. The directory was first seen late on 20 November 2026, twenty days after CVE-2026-119977 was published in NVD, and all 47 files were indexed on 21 November 2026.

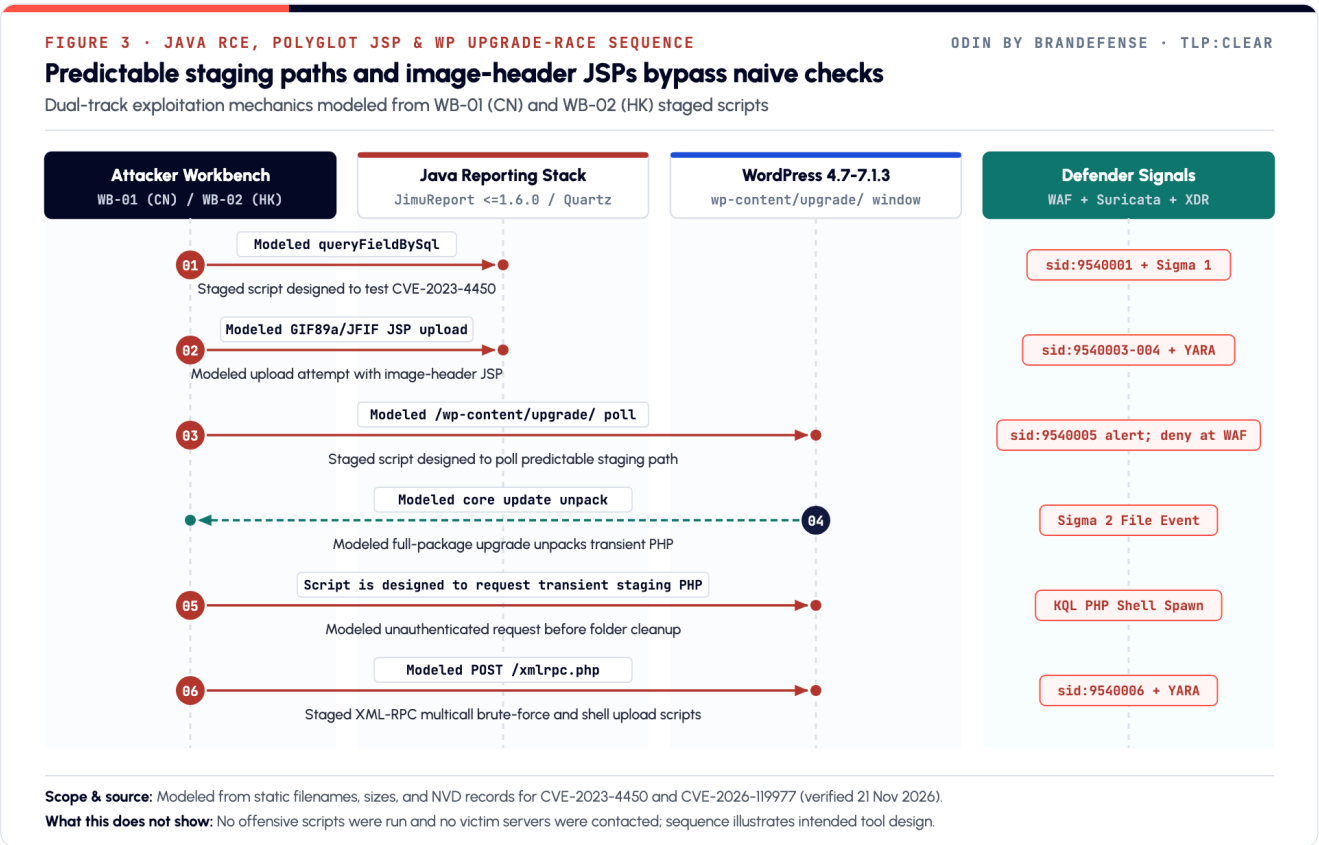


Figure 3: Predictable staging paths and image-header JSPs bypass naive checks

Racing predictable staging paths in `CVE-2026-119977`

The standout file on Workbench B is a 4,377-byte Python upgrade-race script (filename withheld). It is built to exploit **CVE-2026-119977**, an unauthenticated remote code execution vulnerability in WordPress Core (T1190).

CVE-2026-119977 affects **WordPress Core versions 4.7 through 7.1.3** across 25 release branches. NVD published the record on 31 October 2026 with a CVSS v3.1 score of 8.1 High (verified 21 November 2026). During a full-package core upgrade, WordPress extracts the new archive into a staging directory under `wp-content/upgrade/` inside the web root. It then copies the files into place and removes the temporary folder.

In vulnerable versions, those staging paths are predictable and reachable over HTTP while the update is running. An attacker cannot trigger the upgrade without credentials, which is why CVSS rates attack complexity as High. Instead, the staged race script is designed to use concurrent worker threads to poll predictable `/wp-content/upgrade/` paths and request newly unpacked PHP files during the brief upgrade window.

WordPress fixed CVE-2026-119977 in **7.1.4**, **7.0.8**, **6.9.11**, **6.8.12**, **6.7.11**, and **6.6.11**. It also shipped security point updates across older branches from **4.7.39** through **6.5.14**. Blocking all external HTTP requests to `/wp-content/upgrade/` at the reverse proxy or WAF closes the race window even before patching finishes.

Fallback CMS exploits, XML-RPC brute-forcing, and pivot helpers

Core upgrades happen infrequently on a single site. Workbench B also holds 44 other files, an open-source administration archive, and a 9-byte HTTP error stub:

Tool group on Workbench B	Files	Sampled file sizes (filenames withheld)	Staged capability
Upgrade race and CMS PoC	3	Python race script (4,377 B), Python CMS PoC (84,986 B), JSON theme catalog (25,020 B)	Targets core upgrade paths and tests vulnerable WordPress themes
Webshell deployers	2	Two Python uploader scripts (2,172 B and 1,374 B)	Designed to upload PHP webshells through admin plugin or theme routes (T1505.003)
Login and XML-RPC brute force	9	Python login brute-forcer (2,056 B), XML-RPC multicall script (1,327 B), user enumerator (1,015 B), plus 6 helpers	Enumerates users and amplifies password guesses via `system.multicall` (T1110.001)
Recon and backup scanners	5	Duplicator backup scanner (1,353 B), two directory scanners (1,443 B and 1,786 B), REST summary script (1,009 B), plus 1 helper	Checks for exposed Duplicator backup archives, REST user lists, and directories
FTP, PHP wrappers, and network pivots	18	FTP test script (717 B), two PHP wrapper scripts (718 B and 294 B), redirect server (1,273 B), TCP forwarder (1,062 B), plus 13 helpers	Tests FTP credentials, PHP `zlib:` / `ftp://` wrappers, HTTP redirects, and TCP forwarding
Wordlists and plugin lists	8	Eight `.txt` candidate lists (829,702 B, 357,076 B, 26,159 B, 941 B, and 4 smaller lists)	Supplies password masks, site-tailored candidates, and plugin probe names

WordPress's `/xmlrpc.php` endpoint supports the `system.multicall` method, which packs hundreds of `wp.getUsersBlogs` login attempts into one HTTP POST request. The staged XML-RPC script is designed to use that feature to bypass naive per-request rate limits while testing candidates from the co-hosted password lists. The companion deployer scripts are written to automate PHP webshell uploads through authenticated admin routes.

05

CHAPTER 05

Workbench C: Call-Center Web-Phone Probes and Enumeration Logs

IN SHORT

Workbench C was observed on 20 November 2026 in China (port withheld). Its four 148 KB text entries cover `/VICIdial/phone`, a portal login probe, an enumeration log, and a backslash probe. Defenders should lock call-center web panels behind VPNs and keep recon log hashes out of blocklists.

05 Workbench C: Call-Center Web-Phone Probes and Enumeration Logs

Workbench C (China, port withheld) is the smallest of the three open directories, reporting 4 exposed files that were all sampled on 20 November 2026. Figure 4 compares the directory layouts across the three servers.



Figure 4: Open directories expose Java RCE scripts, WP race code, and recon logs

All four entries on Workbench C are text files whose sizes cluster within a 68-byte band between 148,129 and 148,197 bytes:

Entry on Workbench C	Size (bytes)	What it represents
Portal login-registration probe (path withheld)	148,197	Portal login and registration probe entry (digest withheld at source)
Backslash path-normalization probe	148,180	Backslash path-normalization probe testing proxy versus origin routing
Web-surface enumeration log (filename withheld)	148,138	Saved CMS and web-surface enumeration output log
`/VICIdial/phone` route entry	148,129	Web-agent softphone route for the VICIdial call-center platform

VICIdial is an open-source Asterisk contact-center suite used by customer-support and telemarketing teams. Attackers scan for internet-exposed `/VICIdial/phone` and `/agc/` agent portals to abuse SIP trunks for toll fraud, read customer call records, or chain authenticated dialer vulnerabilities.

Because the four files on Workbench C are uniform 148 KB enumeration logs or templated HTTP responses rather than standalone exploit tools, we withhold all Workbench C hashes. Defenders should instead use `/VICIdial/phone` as a network hunt path to find any call-center interface left open to the public internet.

CHAPTER 06

Analytical Model Boundaries and Alternative Explanations

IN SHORT

Zero shared IP addresses or file hashes connect the three directories, so we treat them as an analytical model rather than one group. Because directory ownership is unconfirmed and 164 sampled entries are bundles, stubs, wordlists, or logs, all file hashes are withheld. Defenders should hunt on request and process behavior.

06 Analytical Model Boundaries and Alternative Explanations

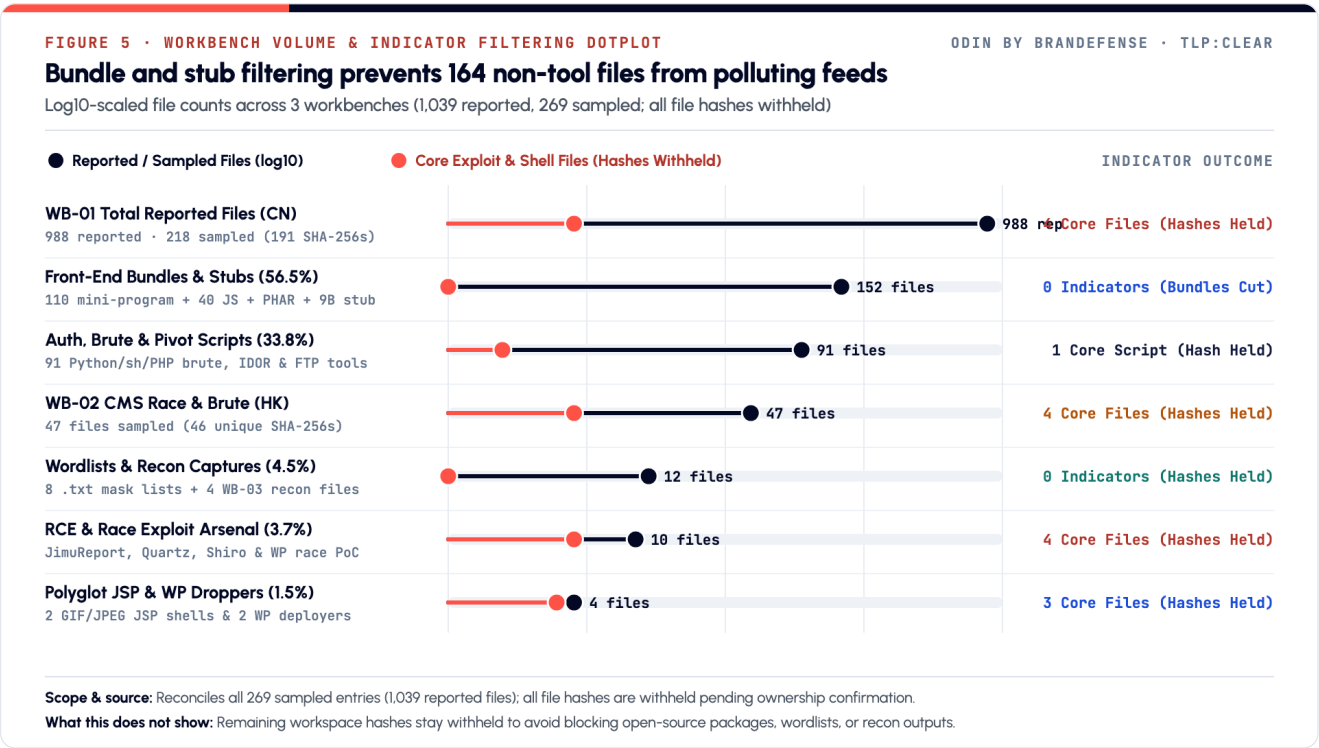


Figure 5: Bundle and stub filtering prevents 164 non-tool files from polluting feeds

Four plain analytical boundaries apply to everything in this study:

1. **Three separate hosts, not one proven actor.** Workbenches A, B, and C were recorded between 13 and 21 November 2026 and focus on web and middleware targets in Asia-Pacific. However, zero shared hosts, zero shared SHA-256 digests, and zero shared filenames connect them. Grouping them is an analytical model for defenders running hybrid Java and PHP stacks, not an attribution claim.
2. **Staged scripts do not confirm breaches.** Finding JimuReport scanners, polyglot JSP webshells, or WordPress upgrade-race scripts on an open server shows what the operator prepared. Nothing in our dataset confirms whether any target JimuReport, Quartz, WordPress, or VICIdial system was breached.
3. **Client bundles and stubs are noise.** Over three-fifths of the sampled entries (164 of 269, 61.0%) are unpacked client JavaScript bundles (150 files), the WordPress CLI PHAR archive (1 file), and the 9-byte `Not Found` stub (1 file). The rest of those 164 files are password wordlists (8 files) and Workbench C recon captures (4 files). Keeping all 164 out of hash feeds protects SOC teams from self-inflicted false alerts.
4. **An exposed security-testing workspace is plausible on Workbench A.** Test-style script roles on Workbench A (covering CAPTCHA reuse, configuration checks, and OAuth grant tests; filenames withheld) could fit an application-security consultant who left a testing directory open (port withheld). Even so, leaving live RCE scanners and polyglot JSP webshells on unauthenticated HTTP creates real risk, and all host addresses, ports, and file hashes stay withheld.

07

CHAPTER 07

What Defenders Should Do

IN SHORT

This chapter lists the actions defenders should take, in priority order.

07 What Defenders Should Do

Java reporting and middleware

- [ ] Upgrade JeecgBoot JimuReport to version 1.6.1 or later across every environment to close CVE-2023-4450.
- [ ] Block direct internet access to `/jmreport/*`, `/sys/quartzJob/*`, `/scheduler/jobs`, and Nacos admin endpoints at the reverse proxy, requiring strong authentication and role checks.
- [ ] Rotate default or static Apache Shiro `RememberMe` encryption keys and update Fastjson libraries to safe versions with `safeMode` enabled.

WordPress and PHP platforms

- [ ] Update WordPress Core to 7.1.4, 7.0.8, 6.9.11, 6.8.12, 6.7.11, 6.6.11, or the fixed maintenance release for your branch to remediate CVE-2026-119977.
- [ ] Deny all external HTTP requests to `/wp-content/upgrade/` at Nginx, Apache, and the WAF so no caller can reach transient files during core, plugin, or theme updates.
- [ ] Disable `/xmlrpc.php` if Jetpack or mobile RPC clients are not used, or block `system.multicall` at the WAF. Remove leftover Duplicator backup archives and set `allow_url_include = Off` in `php.ini`.

File upload pipelines

- [ ] Store user-uploaded images and documents in object storage or outside Java servlet and PHP execution roots, and turn off script execution on all upload directories.
- [ ] Validate uploads with an explicit extension allowlist, generate random server-side filenames, and re-encode GIF and JPEG images so appended `<%` JSP or PHP tags are stripped.

Network and call-center portals

- [ ] Place VICIdial agent interfaces (`/VICIdial/phone` and `/agc/`) behind corporate VPNs or IP allowlists, and audit staging VPS instances for open HTTP directory listings.

CHAPTER 08

Detection Rules

IN SHORT

This chapter describes each published rule and where to deploy it.

INSIDE THIS CHAPTER

How the Suricata signatures work 24

How the endpoint and file rules work 25

MITRE ATT&CK mapping 25

08 Detection Rules

The full rules are in Appendix B. All four published rule files pass static syntax and lint checks; they were not replayed against live network traffic or running engines on our build host. Test and tune them in your own environment before turning on inline blocking or automated host isolation.

Rule file	What it catches
`odin_jimureport_polyglot_jsp_and_wp_race_exploits.rules`	Suricata 7: eight signatures for JimuReport CVE-2023-4450, Quartz job RCE, polyglot JSP uploads and hits, WordPress `/wp-content/upgrade/` race requests, XML-RPC `system.multicall`, `/VICIdial/phone` recon, and exposed workbench listings
`sentinel_jimureport_polyglot_jsp_and_wp_race_hunt.kql`	Microsoft Sentinel and Defender XDR: Java or PHP workers spawning shells or Python exploit command lines, with matching network calls shown only as context
`sigma_java_reporting_rce_polyglot_jsp_and_wp_race.yml`	Sigma (3 Linux rules): `java` or `php-fpm` spawning shells, `.php` or `.jsp` creation in upgrade or upload paths, and Python command lines targeting product routes
`odin_polyglot_jsp_jimureport_and_wp_race_artifacts.yar`	YARA (4 rules): GIF/JPEG-header JSP polyglots, JimuReport and Quartz RCE scripts, WordPress CVE-2026-119977 race scripts, and XML-RPC/deployer tools

How the Suricata signatures work

SID	Looks for	Why
9540001	Inbound `POST` to `/jmreport/queryFieldBySql` carrying FreeMarker `Execute`, `?new()`, or `java.lang.Runtime`	Catches JimuReport CVE-2023-4450 template injection attempts (block candidate)
9540002	Inbound `POST` to Quartz job routes containing `jobClassName` with `Runtime`, `ProcessBuilder`, `InitialContext`, or `GroovyShell`	Catches Java Quartz scheduler command-execution probes
9540003	Inbound `POST` body naming `.jsp` that pairs `GIF89a`, `GIF87a`, or JPEG `JFIF` magic bytes with `<%` JSP tags	Catches multipart uploads of image-header polyglot JSP webshells (block candidate)
9540004	Inbound request URI targeting `.jsp` files inside `/upload/`, `/uploads/`, `/files/`, `/media/`, or `/static/`	Catches direct HTTP access to uploaded JSP webshells in media folders (block candidate)
9540005	Inbound request URI targeting `.php` files inside `/wp-content/upgrade/`	Catches CVE-2026-119977 upgrade-window race requests; visitors never need this path (block candidate)
9540006	Inbound `POST` to `/xmlrpc.php` containing `system.multicall` and `wp.getUsersBlogs`	Hunt or rate-limit signal for amplified WordPress password guessing
9540007	Inbound request URI matching `/VICIdial/phone`	Hunt signal for external reconnaissance against call-center agent portals
9540008	Inbound HTTP 200 `Directory listing for /` referencing `CVE-2026-119977`, `CVE-2023-4450`, `jmreport`, or `xmlrpc`	Flags exposed exploit workbenches during perimeter monitoring

How the endpoint and file rules work

The Sentinel KQL query looks back 7 days and joins `DeviceProcessEvents` with `DeviceNetworkEvents`. On the process side, it flags `java`, `php-fpm`, `apache2`, `httpd`, or `w3wp.exe` starting shells or recon binaries. It also checks for Python or `curl` command lines referencing `/jmreport/queryFieldBySql`, `/sys/quartzJob`, `/wp-content/upgrade/`, or `system.multicall`. Network filtering uses `contains` on `RemoteUrl` for `/jmreport/queryFieldBySql`, `/wp-content/upgrade/`, `/xmlrpc.php`, and `/VICIdial/phone`. Use HTTP or WAF logs where `RemoteUrl` lacks a path, and keep exact hostnames in the named allowlists.

The three Linux Sigma rules cover `auditd` and eBPF telemetry. Rule 1 alerts when `java` or PHP web workers spawn shells outside named build or test servers. Rule 2 watches file creation for `.jsp` files in upload folders or `.php` files under `/wp-content/upgrade/` outside a named runner. Rule 3 catches Python command lines targeting `/jmreport/queryFieldBySql`, `/sys/quartzJob`, `/wp-content/upgrade/`, `system.multicall`, or `/VICIdial/phone`.

YARA rule	Fires when
`ODIN_Polyglot_Image_Header_JSP_Webshell`	A file under 256 KB starts at byte 0 with `GIF89a`, `GIF87a`, or `FF D8 FF` and also contains `<%` plus a Java execution API
`ODIN_JimuReport_and_Quartz_RCE_Scanner_Script`	A script under 128 KB combines JimuReport SQL/show routes or Quartz `jobClassName` routes with FreeMarker or Java execution calls
`ODIN_WordPress_CVE_2026_119977_Upgrade_Race_Exploit`	A script under 256 KB combines `/wp-content/upgrade/` with Python threading/async pools and `CVE-2026-119977` or core-update markers
`ODIN_WordPress_XMLRPC_Multicall_and_Shell_Deployer`	A script under 256 KB combines `/xmlrpc.php`, `system.multicall`, and `wp.getUsersBlogs`, or WordPress admin theme/plugin upload paths

MITRE ATT&CK mapping

Tactic	Technique	How the tooling uses it	Detection coverage
Initial Access	T1190 Exploit Public-Facing Application	JimuReport (CVE-2023-4450), Quartz job, and WordPress upgrade-race (CVE-2026-119977) scripts	SIDs 9540001, 9540002, 9540005; Sigma rules 1 to 3; YARA
Persistence	T1505.003 Server Software Component: Web Shell	GIF and JPEG polyglot JSP webshells and WordPress PHP shell upload scripts	SIDs 9540003, 9540004; KQL; Sigma rules 1 and 2; YARA
Credential Access	T1110.001 Brute Force: Password Guessing	WordPress login brute-force, XML-RPC `system.multicall`, and Shiro key scripts	SID 9540006; Sigma rule 3; YARA
Reconnaissance	T1595.002 Active Scanning: Vulnerability Scanning	`/VICIdial/phone` probing plus directory and backup scanners	SIDs 9540007, 9540008; KQL

09

CHAPTER 09

Indicators

IN SHORT

This chapter lists the published indicators and explains what stays withheld.

INSIDE THIS CHAPTER

File hashes withheld pending ownership confirmation 27

Behavioral request and path patterns (report only) 27

09 Indicators

Appendix A holds 0 published network addresses and 0 file hashes because directory ownership is not established. Six generic product routes and RPC patterns appear in this report for WAF rules, reverse proxies, and access-log hunts.

Type	Count	Where to use it
Network addresses	0	All three open-directory host addresses and ports are withheld
File hashes (SHA-256)	0	All file hashes are withheld pending ownership confirmation
Product routes and RPC patterns	6	Listed below for WAF rules, reverse proxies, and access-log hunts

File hashes withheld pending ownership confirmation

All SHA-256 file hashes from Workbenches A, B, and C are withheld because ownership of the exposed servers is not established. This includes the eight core exploit and webshell files as well as the 164 client bundle, PHAR, stub, wordlist, and reconnaissance files.

Behavioral request and path patterns (report only)

Pattern	Type	How to use it
`/jmreport/queryFieldBySql`	HTTP URI route	Restrict externally and inspect POST bodies for FreeMarker expressions
`/jmreport/show`	HTTP URI route	Restrict externally and inspect requests for template injection expressions
`/sys/quartzJob/`	HTTP URI route	Restrict externally and audit job-registration requests for `jobClassName`
`/wp-content/upgrade/`	Staging path prefix	Block all external HTTP requests at the web server and WAF
`system.multicall` on `/xmlrpc.php`	XML-RPC method	Block or rate-limit when XML-RPC batch authentication is unneeded
`/VICIdial/phone`	Call-center URI	Restrict to internal agent VPNs and hunt external hits

Withheld: all three open-directory IP addresses and ports, all workbench-specific file paths and script names, and the unpacked mini-program AppID. All 240 unique SHA-256 digests from the sampled directories are also withheld.

10

CHAPTER 10

Scope and Limits

IN SHORT

This chapter sets out what the evidence can show and what it cannot.

10 Scope and Limits

- **Observation window.** All findings come from directory samples recorded between 13 and 21 November 2026.
- **Static files, not target telemetry.** Our review covers directory listings, file sizes, paths, and SHA-256 digests. Nothing we observed shows whether these scripts were used against or breached any specific downstream organization.
- **Analytical model, not a single group.** Zero shared hosts, hashes, or filenames link Workbenches A, B, and C. We group them only to illustrate complementary threats to Java middleware and PHP CMS stacks.
- **Assessment workspace possibility.** Structured test script names on Workbench A could belong to a security assessment team that misconfigured a server, which is one reason all host addresses stay withheld.
- **Strict hash filtering.** Hashes of downloaded front-end bundles, open-source CLI archives, wordlists, and enumeration logs are withheld so they cannot cause false positives or expose target material.
- **Rule validation.** The Suricata, Sentinel KQL, Sigma, and YARA files were checked statically on 21 November 2026 and were not replayed against live traffic, so tune them locally before production enforcement.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. National Vulnerability Database (NIST). CVE-2023-4450 Detail: JeecgBoot JimuReport Template Handler Injection. https://nvd.nist.gov/vuln/detail/CVE-2023-4450 (verified 21 November 2026)
2. National Vulnerability Database (NIST). CVE-2026-119977 Detail: WordPress Core Upgrade Staging Path Remote Code Execution. https://nvd.nist.gov/vuln/detail/CVE-2026-119977 (verified 21 November 2026)
3. JeecgBoot Project. JimuReport Releases and Security Updates (v1.6.1+). https://github.com/jeecgboot/JimuReport/releases (verified 21 November 2026)
4. Quartz Enterprise Job Scheduler. Official Documentation. https://www.quartz-scheduler.org/documentation/ (verified 21 November 2026)
5. OWASP Cheat Sheet Series. File Upload Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html (verified 21 November 2026)
6. WordPress Documentation. Updating WordPress. https://wordpress.org/documentation/article/updating-wordpress/ (verified 21 November 2026)
7. MITRE ATT&CK. T1190 Exploit Public-Facing Application. https://attack.mitre.org/techniques/T1190/ (verified 21 November 2026)
8. MITRE ATT&CK. T1505.003 Server Software Component: Web Shell. https://attack.mitre.org/techniques/T1505/003/ (verified 21 November 2026)

APPENDICES

Indicators and Detection Rules

FOR DEFENDERS

No individual indicators are published for this study; Appendix A explains why. Appendix B carries every published detection rule in full.

INSIDE

A. Indicators of compromise (0) 33

B. Detection rules (4 files) 34

A Indicators of Compromise

No individual addresses or file hashes are published for this study. The hosts in this report are exposed or unattributed systems rather than confirmed adversary infrastructure, so they stay anonymous. Use the detection rules in Appendix B.

B Detection Rules

Each rule file loads in its own engine (Suricata 7, YARA-X, Kusto, pySigma). Tune thresholds and allow-lists to your environment before alerting or blocking. The same files can be downloaded from the online edition.

odin_jimureport_polyglot_jsp_and_wp_race_exploits.rules

SURICATA

```
# Odin by Brandefense - Monograph 40 / 46 Suricata 7.x Detection Suite
# Coverage: JeecgBoot JimuReport CVE-2023-4450 RCE, Java Quartz job scheduler RCE probes,
#           GIF/JPEG polyglot JSP webshell uploads and URI hits, WordPress CVE-2026-119977
#           upgrade-window staging access, XML-RPC multicall brute force, VICIdial probes,
#           and open-directory exploit workbench listings
# Classification: TLP:CLEAR

alert http any any -> $HOME_NET any (msg:"ODIN [R40] JeecgBoot JimuReport queryFieldBySql Template Injection RCE
Attempt (CVE-2023-4450)"; flow:established,to_server; http.method; content:"POST"; http.uri; bsize:>24;
content:"/jmreport/queryFieldBySql"; fast_pattern; http.request_body; bsize:>16; pcre:"/
(freemarker\.template\.utility\.Execute|\?new\s*\(\s*\)|java\.lang\.Runtime)/i"; classtype:web-application-
attack; sid:9540001; rev:1; metadata:created_at 2026_11_21, cve CVE_2023_4450, deployment perimeter_and_waf,
tactic initial_access;)

alert http any any -> $HOME_NET any (msg:"ODIN [R40] Java Quartz Scheduler Job Class Remote Command Execution
Probe"; flow:established,to_server; http.method; content:"POST"; http.uri; bsize:>10; pcre:"/\/(sys\/quartzJob|
quartz\/job|scheduler\/jobs)/i"; http.request_body; bsize:>18; content:"jobClassName"; fast_pattern; pcre:"/
(java\.lang\.Runtime|ProcessBuilder|javax\.naming\.InitialContext|GroovyShell)/i"; classtype:web-application-
attack; sid:9540002; rev:1; metadata:created_at 2026_11_21, deployment perimeter_and_waf, tactic execution;)

alert http any any -> $HOME_NET any (msg:"ODIN [R40] Image-Header Polyglot JSP Webshell Upload (GIF or JPEG Magic
Bytes With JSP Tag)"; flow:established,to_server; http.method; content:"POST"; http.request_body; bsize:>24;
content:".jsp"; fast_pattern; pcre:"/(GIF89a|GIF87a|\xff\xd8\xff\xe0.{0,16}JFIF)[\s\S]{0,512}<%[@=\s]/";
classtype:web-application-attack; sid:9540003; rev:1; metadata:created_at 2026_11_21, deployment
perimeter_and_waf, tactic persistence;)

alert http any any -> $HOME_NET any (msg:"ODIN [R40] Uploaded JSP Webshell Direct Access in Upload or Media
Path"; flow:established,to_server; http.uri; bsize:>10; content:".jsp"; fast_pattern; pcre:"/\/(upload|uploads|
files|media|static)\/[^\r\n]*\.jsp(\?|$)/i"; classtype:web-application-attack; sid:9540004; rev:1;
metadata:created_at 2026_11_21, deployment perimeter_and_waf, tactic command_and_control;)

alert http any any -> $HOME_NET any (msg:"ODIN [R40] WordPress Core Upgrade Staging Path Direct Access
(CVE-2026-119977)"; flow:established,to_server; http.uri; bsize:>20; content:"/wp-content/upgrade/";
fast_pattern; pcre:"/\/wp-content\/upgrade\/[^\/]+\/.*\.php/i"; classtype:web-application-attack; sid:9540005;
rev:1; metadata:created_at 2026_11_21, cve CVE_2026_119977, deployment perimeter_and_waf, tactic initial_access;)

alert http any any -> $HOME_NET any (msg:"ODIN [R40] HUNT WordPress XML-RPC system.multicall Batch Authentication
Attempt"; flow:established,to_server; http.method; content:"POST"; http.uri; bsize:>10; content:"/xmlrpc.php";
http.request_body; bsize:>32; content:"system.multicall"; fast_pattern; content:"wp.getUsersBlogs";
classtype:attempted-admin; sid:9540006; rev:1; metadata:created_at 2026_11_21, deployment perimeter_and_waf,
policy behavioral_hunt;)

alert http any any -> $HOME_NET any (msg:"ODIN [R40] HUNT VICIdial Call-Center Web Phone Interface Enumeration
Probe"; flow:established,to_server; http.uri; bsize:>14; content:"/VICIdial/phone"; fast_pattern;
classtype:attempted-recon; sid:9540007; rev:1; metadata:created_at 2026_11_21, deployment perimeter_and_waf,
tactic reconnaissance;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R40] Open-Directory Java RCE or WordPress Race
Workbench Listing"; flow:established,from_server; http.stat_code; content:"200"; http.response_body; bsize:>32;
content:"Directory listing for /"; fast_pattern; pcre:"/(CVE-2026-119977|CVE-2023-4450|jmreport|xmlrpc)/i";
classtype:attempted-recon; sid:9540008; rev:1; metadata:created_at 2026_11_21, deployment perimeter, tactic
resource_development;)
```

odin_polyglot_jsp_jimureport_and_wp_race_artifacts.yar

YARA

```
/*
   Odin by Brandefense - Monograph 40 / 46 YARA Artifact Suite
   Scope: Structural detection rules for GIF/JPEG image-header polyglot JSP webshells,
          JeecgBoot JimuReport (CVE-2023-4450) and Java Quartz RCE scanners,
          WordPress Core CVE-2026-119977 upgrade-window race scripts, and
          WordPress XML-RPC multicall / shell deployment scripts.
   Classification: TLP:CLEAR
*/

rule ODIN_Polyglot_Image_Header_JSP_Webshell
{
    meta:
        author = "Odin by Brandefense"
        monograph = "40 / 46"
        classification = "TLP:CLEAR"
        action = "BLOCK_POLYGLOT_JSP_WEBSHELL"
        description = "Matches JSP webshells prepending GIF87a/GIF89a or JPEG magic bytes ahead of JSP tags and
Java execution primitives"
    strings:
        $gif1 = "GIF89a" ascii
        $gif2 = "GIF87a" ascii
        $jpg1 = { FF D8 FF }
        $jsp1 = "<%" ascii
        $ex1 = "getRuntime" ascii
        $ex2 = "ProcessBuilder" ascii
        $ex3 = "defineClass" ascii
        $ex4 = "javax.crypto.Cipher" ascii
    condition:
        filesize < 256KB and ($gif1 at 0 or $gif2 at 0 or $jpg1 at 0) and $jsp1 and 1 of ($ex*)
}

rule ODIN_JimuReport_and_Quartz_RCE_Scanner_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "40 / 46"
        classification = "TLP:CLEAR"
        action = "HUNT_JAVA_MIDDLEWARE_RCE_SCANNER"
        description = "Matches Python scanners probing JeecgBoot JimuReport CVE-2023-4450 template injection or
Java Quartz scheduler RCE"
    strings:
        $jmr1 = "/jmreport/queryFieldBySql" ascii
        $jmr2 = "/jmreport/show" ascii
        $ftl1 = "freemarker.template.utility.Execute" ascii
        $qtz1 = "jobClassName" ascii
        $qtz2 = "sys/quartzJob" ascii
        $pay1 = "getRuntime" ascii
        $pay2 = "ProcessBuilder" ascii
        $pay3 = "InitialContext" ascii
    condition:
        filesize < 128KB and ((($jmr1 or $jmr2) and ($ftl1 or 1 of ($pay*))) or (($qtz1 or $qtz2) and 1 of
($pay*)))
}

rule ODIN_WordPress_CVE_2026_119977_Upgrade_Race_Exploit
{
    meta:
        author = "Odin by Brandefense"
        monograph = "40 / 46"
        classification = "TLP:CLEAR"
        action = "HUNT_WP_UPGRADE_RACE_EXPLOIT"
        description = "Matches Python race-condition scripts polling predictable WordPress wp-content/upgrade
staging paths during CVE-2026-119977 windows"
    strings:
        $up1 = "/wp-content/upgrade/" ascii
        $rc1 = "ThreadPoolExecutor" ascii
        $rc2 = "threading.Thread" ascii
        $rc3 = "asyncio.gather" ascii
```

```
        $ctx1 = "CVE-2026-119977" ascii nocase
        $ctx2 = "update-core.php" ascii
        $ctx3 = "wordpress-7." ascii nocase
    condition:
        filesize < 256KB and $up1 and 1 of ($rc*) and 1 of ($ctx*)
}

rule ODIN_WordPress_XMLRPC_Multicall_and_Shell_Deployer
{
    meta:
        author = "Odin by Brandefense"
        monograph = "40 / 46"
        classification = "TLP:CLEAR"
        action = "HUNT_CMS_BRUTE_AND_DEPLOYER"
        description = "Matches Python WordPress XML-RPC system.multicall brute-force scripts and theme/plugin
webshell uploaders"
    strings:
        $x1 = "/xmlrpc.php" ascii
        $x2 = "system.multicall" ascii
        $x3 = "wp.getUsersBlogs" ascii
        $d1 = "wp-admin/theme-editor.php" ascii
        $d2 = "wp-admin/update.php?action=upload-plugin" ascii
        $d3 = "application/zip" ascii
    condition:
        filesize < 256KB and (($x1 and $x2 and $x3) or (($d1 or $d2) and $d3))
}
```

sentinel_jimureport_polyglot_jsp_and_wp_race_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Monograph 40 / 46 Microsoft Sentinel & Defender XDR Hunt
// Correlates Java servlet/reporting and PHP-FPM child shell execution, attacker RCE/race
// script invocations, and HTTP endpoints for JimuReport CVE-2023-4450, polyglot JSP shells,
// and WordPress Core CVE-2026-119977 upgrade-window race attempts
// Classification: TLP:CLEAR

let Lookback = 7d;
let ApprovedAdminHosts = dynamic(["CMS-UPGRADE-RUNNER-01", "JAVA-REPORT-BUILDER-01"]);
let ApprovedSecurityHosts = dynamic(["SEC-VALIDATION-LNX-01", "SEC-VALIDATION-WEB-01"]);
let ProcSignals =
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAdminHosts) or DeviceName in~ (ApprovedSecurityHosts))
    | where (
        InitiatingProcessFileName in~ ("java", "java.exe", "php-fpm", "php-fpm8.2", "php-fpm8.3", "php-cgi",
"apache2", "httpd", "w3wp.exe")
        and FileName in~ ("sh", "bash", "dash", "cmd.exe", "powershell.exe", "id", "whoami", "uname", "curl",
"wget")
    )
    or (
        FileName in~ ("python", "python3", "python.exe", "curl", "curl.exe")
        and (
            ProcessCommandLine contains "/jmreport/queryFieldBySql"
            or ProcessCommandLine contains "/sys/quartzJob"
            or ProcessCommandLine contains "/wp-content/upgrade/"
            or ProcessCommandLine contains "system.multicall"
        )
    )
    | project ProcTime = Timestamp, DeviceId, DeviceName, AccountName, InitiatingProcessFileName, FileName,
ProcessCommandLine, SHA256;
let NetSignals =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAdminHosts) or DeviceName in~ (ApprovedSecurityHosts))
    | where InitiatingProcessFileName in~ ("python", "python3", "curl", "java", "php-fpm", "apache2", "httpd",
"nginx")
    | where RemoteUrl contains "/jmreport/queryFieldBySql" or RemoteUrl contains "/wp-content/upgrade/" or
RemoteUrl contains "/xmlrpc.php" or RemoteUrl contains "/VICIdial/phone"
    | summarize NetHits = count(), SampleUrls = make_set(RemoteUrl, 5), RemoteIPs = make_set(RemoteIP, 5) by
```

```
DeviceId;
ProcSignals
| join kind=leftouter NetSignals on DeviceId
| extend ResponsePriority = case(
    ProcessCommandLine contains "/jmreport/queryFieldBySql" or ProcessCommandLine contains "/sys/quartzJob" or
ProcessCommandLine contains "/wp-content/upgrade/" or ProcessCommandLine contains "system.multicall",
"ISOLATE_EXPLOIT_TOOL_EXECUTION",
    InitiatingProcessFileName in~ ("java", "java.exe") and FileName in~ ("sh", "bash", "dash", "cmd.exe", "id",
"whoami"), "ISOLATE_JAVA_RCE_OR_JSP_SHELL",
    "ISOLATE_CMS_WORKER_SHELL_SPAWN"
)
| project ProcTime, DeviceName, AccountName, ResponsePriority, InitiatingProcessFileName, FileName,
ProcessCommandLine, SHA256, NetHits, SampleUrls, RemoteIPs
| sort by ProcTime desc
```

sigma_java_reporting_rce_polyglot_jsp_and_wp_race.yml

SIGMA

```
title: Java Reporting Container or PHP Worker Spawning Shell Subprocess
id: 4b91e2a0-7d34-4f18-9c62-202611214001
status: experimental
description: Detects Java servlet/reporting runtimes (JeecgBoot JimuReport, Quartz, Tomcat) or PHP web workers
spawning OS shell or reconnaissance binaries consistent with CVE-2023-4450 template injection, polyglot JSP
execution, or WordPress CVE-2026-119977 upgrade-race code execution.
author: Odin by Brandefense
date: 2026/11/21
tags:
  - attack.initial-access
  - attack.t1190
  - attack.persistence
  - attack.t1505.003
  - attack.execution
logsource:
  category: process_creation
  product: linux
detection:
  selection_parent:
    ParentImage|endswith:
      - '/java'
      - '/php-fpm'
      - '/php-fpm8.2'
      - '/php-fpm8.3'
      - '/php-cgi'
      - '/apache2'
      - '/httpd'
  selection_child:
    Image|endswith:
      - '/sh'
      - '/bash'
      - '/dash'
      - '/id'
      - '/whoami'
      - '/uname'
      - '/curl'
      - '/wget'
  filter_validation_web:
    ComputerName:
      - 'SEC-VALIDATION-WEB-01'
      - 'JAVA-REPORT-BUILDER-01'
  condition: selection_parent and selection_child and not filter_validation_web
falsepositives:
  - Legacy Java batch or PHP administration scripts configured to invoke external OS utilities on non-hardened
hosts
level: critical
---
title: Polyglot JSP Webshell Drop or WordPress Upgrade Staging PHP File Creation
id: 8c27f5d3-1a49-4e85-b390-202611214002
status: experimental
description: Detects creation of JSP files in web upload or media directories or PHP files written inside
```

```
  WordPress wp-content/upgrade staging directories outside registered CMS release runners.
  author: Odin by Brandefense
  date: 2026/11/21
  tags:
    - attack.persistence
    - attack.t1505.003
    - attack.initial-access
    - attack.t1190
  logsource:
    category: file_event
    product: linux
  detection:
    selection_target:
      TargetFilename|re: '(/wp-content/upgrade/[^/]+/.*\.php$|/(upload|uploads|files|media|static)/.*\.jsp$)'
    filter_upgrade_runner:
      ComputerName:
        - 'CMS-UPGRADE-RUNNER-01'
        - 'SEC-VALIDATION-WEB-01'
    condition: selection_target and not filter_upgrade_runner
  falsepositives:
    - Scheduled WordPress core full-package updates running directly on origin hosts not listed in CMS-UPGRADE-
  RUNNER-01
  level: high
  ---
  title: JimuReport RCE Scanner or WordPress CVE-2026-119977 Race Script Execution
  id: 2f64c8b9-6e12-4c71-a504-202611214003
  status: experimental
  description: Detects Linux command-line execution of Python scripts targeting JeecgBoot JimuReport, Quartz
  scheduler routes, WordPress wp-content/upgrade staging paths, XML-RPC system.multicall, or VICIdial web-phone
  endpoints.
  author: Odin by Brandefense
  date: 2026/11/21
  tags:
    - attack.initial-access
    - attack.t1190
    - attack.credential-access
    - attack.t1110.001
  logsource:
    category: process_creation
    product: linux
  detection:
    selection_runtime:
      Image|endswith:
        - '/python'
        - '/python3'
    selection_cli:
      CommandLine|contains:
        - '/jmreport/queryFieldBySql'
        - '/sys/quartzJob'
        - '/wp-content/upgrade/'
        - 'system.multicall'
        - '/VICIdial/phone'
    filter_security_host:
      ComputerName:
        - 'SEC-VALIDATION-LNX-01'
    condition: selection_runtime and selection_cli and not filter_security_host
  falsepositives:
    - Authorized security validation hosts running purple-team exploit simulations
  level: high
```

ODIN

About Odin Research

Odin Research is the public research series of Odin by Brandefense. It documents adversary infrastructure seen on the internet and turns each finding into indicators and detection rules that defenders can use. Victims and third parties stay anonymous, addresses appear only for confirmed adversary infrastructure, and no working attack code is published.

Odin is the agentic analyst inside the Brandefense threat intelligence platform, which unifies Cyber Threat Intelligence, Digital Risk Protection and External Attack Surface Management.

Online edition, indicators and rules · brandefense.io/odin-research/reports/jimureport-probes-polyglot-jsps-and-wordpress-race-tooling/
Short Read · brandefense.io/odin-research/blog/inside-exposed-toolkits-targeting-jimureport-file-uploads-and-wordpress-upgrades/
Defense Kit · brandefense.io/odin-research/defense-kit/