FULL REPORT · ENTERPRISE EXPLOITATION · MONOGRAPH 56 / 64 · 18 MIN READ

Exposed Java Workbench: Heap Captures, FFmpeg Callbacks, and STOMP Test Files

Inside a 225-entry Singapore workbench combining Spring Boot heap captures, CFR Java decompilation, FFmpeg outbound callbacks, and staged STOMP-over-WebSocket scripts

BY
Odin by Brandefense

PUBLISHED
19 November 2026

READING TIME
18 min read

EDITION
Version 1.1 · TLP:CLEAR

SCOPE
Prepared November 2026

1. Executive Summary

IN SHORT

An exposed Java workbench sits on an open directory on a Singapore server recorded on 14–20 November 2026. The workbench reported 208 active files on 20 November and 225 deduplicated entries totaling 193.12 megabytes. It holds Spring Boot heap snapshots, 77 sized Java class files decompiled into 56 sized source files, FFmpeg callback logs, and staged STOMP and JNDI scripts. Block public Actuator routes, restrict FFmpeg network protocols, and authorize STOMP channels.

Enterprise web applications often pair JavaScript frontends with Spring Boot services, media workers, and WebSocket message brokers. Exposed diagnostic routes or permissive media workers can reveal backend logic and internal network paths. This exposed Java workbench brings four distinct Java and media artifact groups onto one server.

Between 14 and 20 November 2026, Odin by Brandefense analyzed an open HTTP directory on a Singapore virtual private server. On 20 November 2026, that directory reported 208 active files and 55 directories.

Across the full 14 to 20 November observation window, deduplicated indexing captured 225 entries. Those entries span 211 sized files totaling 193,118,194 bytes (193.12 megabytes) and 14 unsized

entries across 199 unique digests.

The workbench holds JVM heap snapshots, compiled target Java classes, and private key and hash files. Odin by Brandefense therefore publishes zero server addresses, zero host ports, and zero file SHA-256 digests. All organization names, telecom brands, Java package and class names, scanner domains, client addresses, and captured filenames are withheld.

1. **JVM heap snapshots and frontend bundles expose routes and memory state.** Forty entries (**159.57** megabytes) hold a raw heap capture, an HPROF snapshot, extracted heap strings, route and trace captures, four parsers, and **29** frontend bundle items. The raw heap capture (**117.14** megabytes), HPROF snapshot (**25.74** megabytes), and extracted strings (**14.11** megabytes) sit in root. Together they reveal controller paths, classloaders, and active configuration secrets.

2. **CFR bytecode decompilation rebuilds readable Java source trees.** One hundred forty-six entries cover bytecode extraction and CFR source recovery. An unpacked folder holds **77** sized class files (**82** total), while six output folders hold **56** sized Java source files (**62** total). Compiled Java archives provide no secrecy once bytecode leaks.

3. **Listener logs record FFmpeg and FFprobe client strings.** Eight entries (**2.42** megabytes) pair a Python media server and test MP4 file with HTTP and LDAP listeners. Both listener logs carry internal timestamps dated **13** and **14** October **2026**. Across **2,307** HTTP lines and **2,751** LDAP lines, three lines record `Lavf/58.29.100` and one records `ffprobe`.

4. **Five STOMP scripts and six JNDI files show staged broker preparation.** Eleven entries (**11.69** megabytes) hold five STOMP-over-WebSocket scripts alongside two JNDI archives and four JNDI logs. Legacy flaws such as `CVE-2018-1270` provide architectural context, while staged scripts alone do not confirm remote code execution.

Three Decisions for Enterprise Application Leaders

1. **Block external Spring Boot Actuator routes and rotate exposed secrets.** Disable heapdump, mappings, and trace endpoints in production, block `/actuator/*` at reverse proxies, and rotate credentials after any heap exposure.

2. **Enforce file-only FFmpeg protocol allowlists and isolate worker egress.** Run FFmpeg and FFprobe with strict local protocol allowlists, reject remote playlist URLs, and block outbound internet access from media subnets.
3. **Patch Spring messaging libraries, authorize STOMP topics, and block JNDI egress.** Keep Spring Framework on current branches, enforce per-user checks on SUBSCRIBE and SEND frames, and block outbound LDAP and RMI ports.

2. Workbench Architecture, 225-Entry Reconciliation, and Timeline Scope

The exposed directory sat on a **Singapore commercial virtual private server** (address and port withheld). That server exposed an open HTTP directory alongside a **16,468,015**-byte HTTP directory service log. Security teams use external attack surface management to find exposed diagnostic routes before memory snapshots leak; attack surface management starts with knowing which routes are public.

Odin by Brandefense recorded the directory between `14 and 20 November 2026`. Analysts also apply cyber threat intelligence to map how application-tier artifacts connect across staging hosts. Figure 1 shows the four technical tracks present on the server.

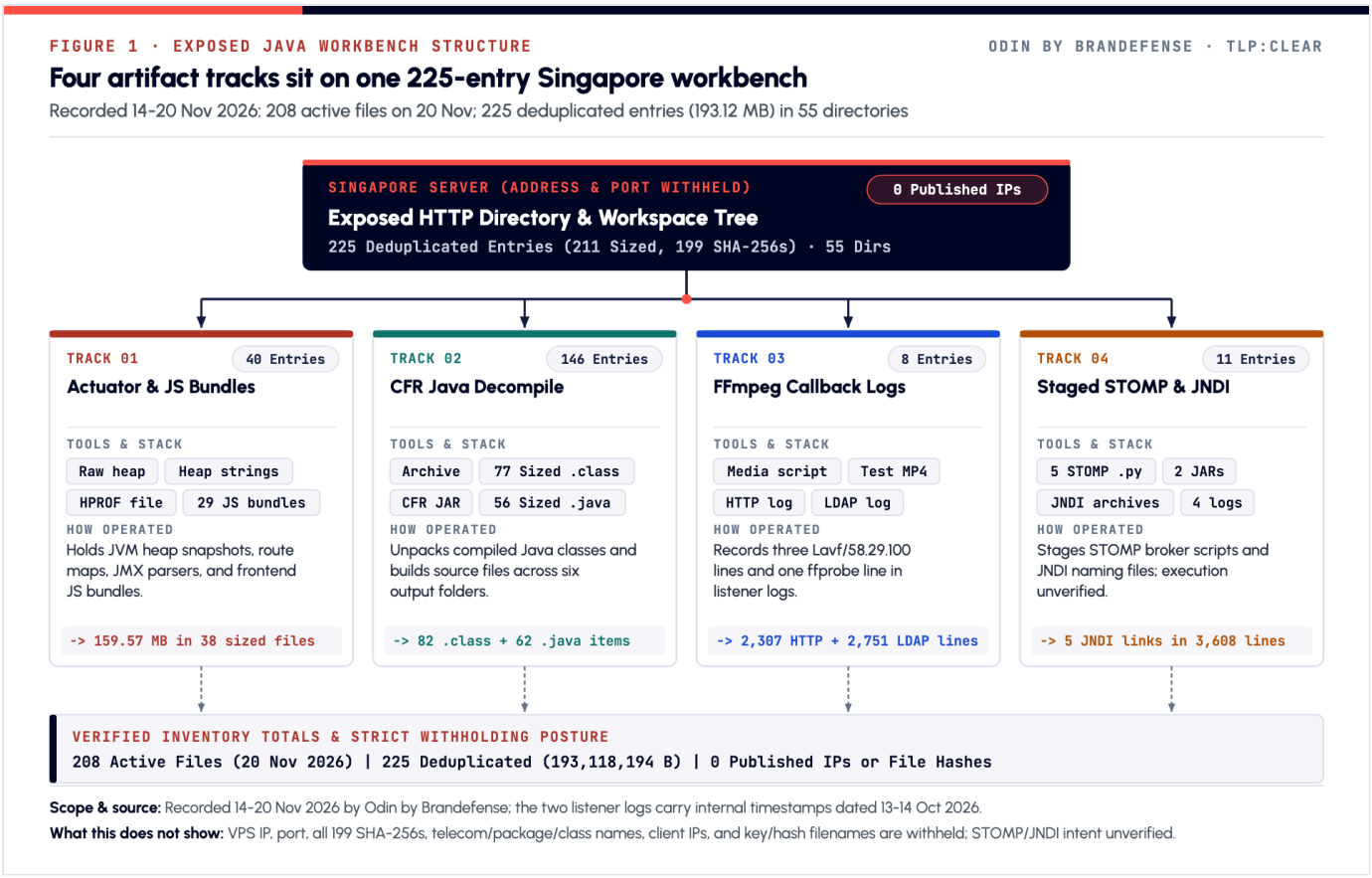

Figure 1: Four artifact tracks sit on one 225-entry Singapore workbench

Reconciling the `20 November` Listing (208 Files) and 225 Deduplicated Entries

Two exact counts describe the server across `14-20 November 2026`. On `20 November 2026`, the live directory listing reported **208 active files** and **55 directories** (**8** top-level folders plus **47** nested subdirectories).

Across the full `14-20 November 2026` observation period, deduplicated indexing captured **225 total entries**. Those **225** entries include **211 sized files** totaling **193,118,194 bytes (193.12 MB)** and **14 unsized entries**. The **208** active files and **225** deduplicated entries are counts from different scopes; the available inventory does not identify which entries account for the difference.

All **225** entries span **199 unique SHA-256 digests**, every one of which is withheld. The table below reconciles the **225** deduplicated entries across the four workspace zones.

Workspace zone	Top-level dirs	Deduplicated entries	Share (%)	Sized files	Unsized entries	Sized bytes	Operational role
Extracted Bytecode Tree	1	82	36.4%	77	5	277,277	Unpacked Java `.class` tree from a 90,453-byte bytecode archive (77 sized `.class` files + 5 unsized; folder, package, and class names withheld).
CFR Decompiled Source	6	62	27.6%	56	6	284,408	Decompiled `.java` source across six output folders: primary folder (48; 43 sized = 218,348 B) plus five subset folders (5, 3, 3, 2, and 1).
Root Workspace (`(root)`)	0 (root)	52	23.1%	51	1	190,819,312	Holds a raw heap capture, an HPROF snapshot, extracted heap strings, route and trace captures, an HTTP service log, a CFR archive, JNDI archives, callback logs, 5 STOMP `.py` scripts, and withheld files.
Frontend JS Bundle Snapshot	1	29	12.9%	27	2	1,737,197	Captured web frontend `.js` bundles (27 sized `.js` files + 1 unsized `.js` + 1 unsized `.txt`; folder and filenames withheld).
Combined deduplicated inventory	**8 top dirs + root**	**225**	**100.0%**	**211**	**14**	**193,118,194**	**Spans 199 withheld SHA-256 digests (208 active files & 55 dirs on `20 Nov 2026`).**

A second breakdown reconciles the same **225** entries (**211** sized, **14** unsized, **193,118,194** bytes) by file type. Text and Java source files (**88**), bytecode binaries (**84**), and scripts (**46**) represent **96.9%** of entries. By byte volume, a raw heap capture (**117.14 MB**) and an HPROF snapshot (**25.74 MB**) make up **74.0%** of the server.

Artifact class (`kind`)	Deduplicated entries	Share (%)	Sized files	Unsized entries	Sized bytes	Verified extensions & contents
Text, source & logs (`text`)	88	39.1%	81	7	35,561,099	**62** `.java` (**56** sized + **6** unsized), **12** `.log`, **5** `.txt`, **2** `.json`, **2** `.pub`, **2** `.conf`, **1** `.out`, **1** `.c`, and **1** 9-byte `.jar` stub.
Bytecode & JAR tools (`exe`)	84	37.3%	79	5	12,797,060	**82** `.class` entries in the extracted bytecode folder (**77** sized = **277,277** B + **5** unsized) and **2** `.jar` archives (**10,357,468** B JNDI archive; **2,162,315** B CFR archive).
Scripts & bundles (`script`)	46	20.4%	45	1	1,783,103	**29** `.js` (**28** in the frontend bundle folder + a **1,586**-byte root `.js` bundle), **13** `.py` (**40,177** B), and **4** `.sh` (**4,740** B).
Raw captures & potfiles (`other`)	2	0.9%	1	1	117,139,838	**1** `.raw` heap capture (**117,139,838** B; filename withheld) and **1** unsized `.pot` entry.
Heap & class archives (`archive`)	2	0.9%	2	0	25,826,510	**1** `.hprof` snapshot (**25,736,057** B; filename withheld) and **1** `.tgz` bytecode archive (**90,453** B; filename withheld).
Private key files (`secret`)	2	0.9%	2	0	3,654	**2** extensionless SSH private keys (**1,843** B and **1,811** B; filenames withheld).
Test media (`media`)	1	0.4%	1	0	6,930	**1** `.mp4` test container (**6,930** B; filename withheld).
Total	**225**	**100.0%**	**211**	**14**	**193,118,194**	**Exact byte-level parity** (193.12 MB).

Figure 5 groups all **225** deduplicated entries into six mutually exclusive clusters (`146 + 40 + 15 + 11 + 8 + 5 = 225`). Decompiled Java classes (**146**) and Actuator heap and bundle files (**40**) account for **82.7%** of all indexed items.

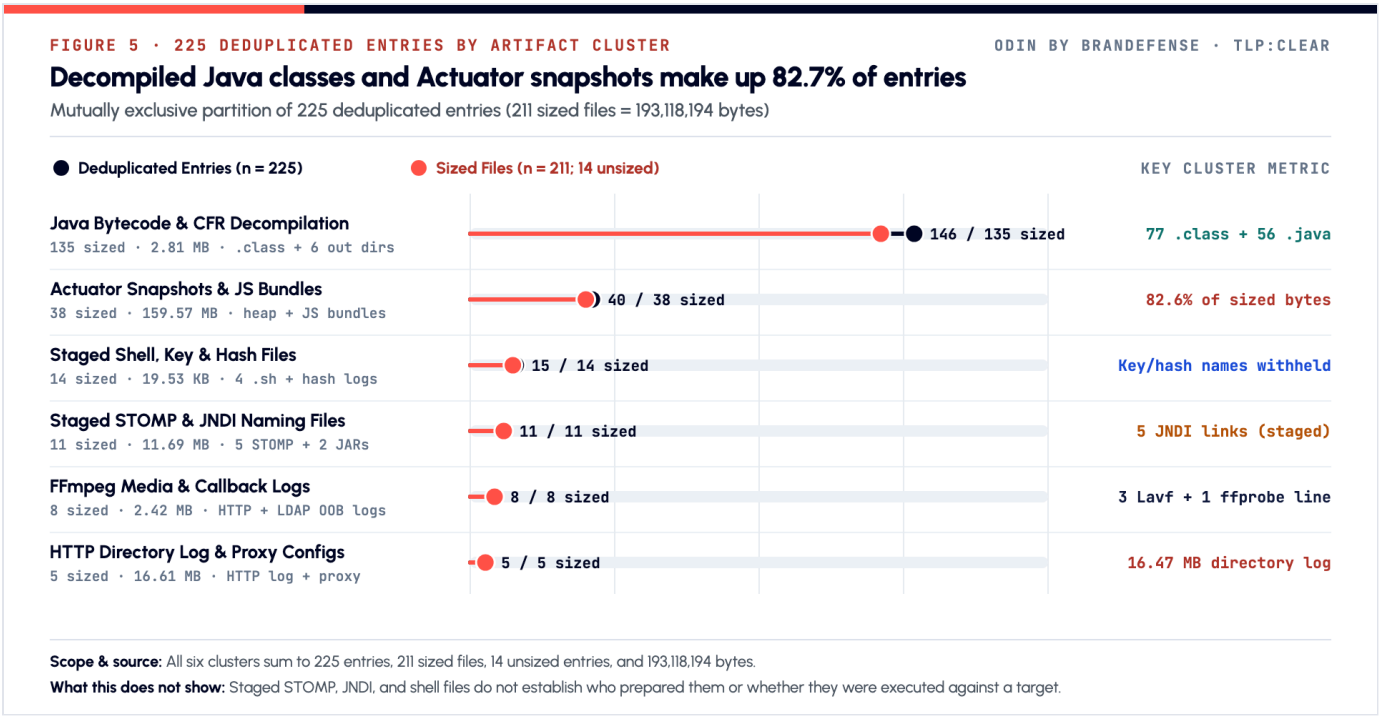

Figure 5: Decompiled Java classes and Actuator snapshots make up 82.7% of entries

Internal Listener Log Dates Versus November Directory Observations

Two distinct date ranges appear in the dataset, and each carries a clear boundary.

Odin by Brandefense observed the open HTTP directory between `14 and 20 November 2026`. That live directory reported **208** active files and **55** directories on `20 November 2026`.

Inside `(root)`, both the HTTP callback log (**399,401** B) and the TCP/LDAP listener log (**564,831** B) carry internal timestamps dated **13** and `14 October 2026`. Those dates show that the logs contain October entries; they do not establish when or where the listeners started. Log byte sizes alone also do not establish continuous runtime.

3. Actuator Heapdump Exposure, CFR Decompilation, and FFmpeg Callbacks

Many enterprise and telecom portals place a single-page JavaScript frontend ahead of Spring Boot microservices and media-transcoding workers. Figure 3 presents an analytical model separating captured snapshots and logged `Lavf/58.29.100` callbacks from staged STOMP and JNDI scripts.

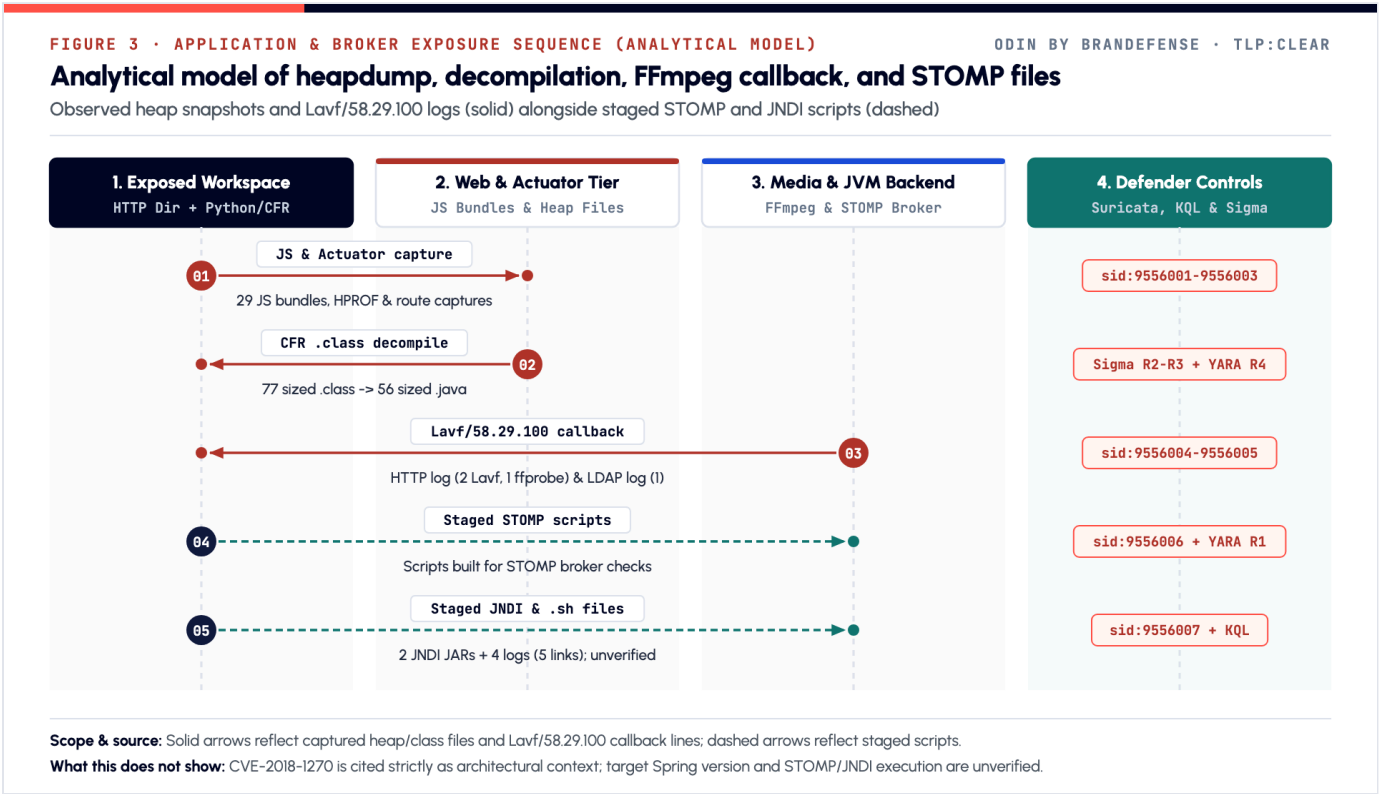

Figure 3: Analytical model of heapdump, decompilation, FFmpeg callback, and STOMP files

Protecting Spring Boot Actuator `/heapdump` Diagnostics and Frontend Bundles (40 Entries)

The Actuator and frontend cluster holds **40** entries (**38** sized files totaling **159,569,751** bytes). Twenty-seven sized `.js` bundles (**1,737,197** B) and two unsized items sit in a frontend bundle folder beside a **1,586**-byte root bundle. Client bundles map REST paths, request headers, and WebSocket broker endpoints (folder, bundle, and brand names withheld).

Root also holds a raw heap capture (**117,139,838** B) and an HPROF snapshot (**25,736,057** B). Two text files hold extracted heap strings (**14,111,011** B) and a secondary heap extract (**799,397** B). Together with route and trace captures and **29** frontend bundle items, they reflect Spring Boot

Actuator's `/actuator/heapdump` route (or legacy `/heapdump`; `[REF-03]`). Live `JAVA PROFILE 1.0.2` memory exports contain active properties, database passwords, session tokens, and signing keys (`[REF-10]`).

Six diagnostic captures and Python helpers sit beside those heap files. Two JSON captures hold route and trace data (**31,992** B and **5,634** B). Four scripts inspect proxy headers (**2,581** B), JMX MBeans (**1,717** B), and classloaders (**1,379** B and **1,362** B). Workbench files alone do not show how the Actuator routes were reached.

Assessing Exposure from Decompiled Java `.class` Archives (146 Entries)

The Java decompilation cluster holds **146** entries (**135** sized files totaling **2,814,453** bytes), making up **64.9%** of the deduplicated inventory. A **90,453**-byte compressed archive unpacks into an extracted bytecode folder holding **82** **`.class`** **entries**. Those entries span **77** sized `.class` files (**277,277** bytes) and **5** unsized entries. All target organization, telecom brand, archive, folder, Java package, and class names are withheld.

Root holds a **2,162,315**-byte CFR Java decompiler archive. CFR generated **62** **`.java`** **entries** (**56** sized `.java` files totaling **284,408** bytes plus **6** unsized entries) across six output folders. Most source files sit in the primary output folder (**48** entries; **43** sized = **218,348** B). Another **14** entries sit across five subset folders (**5**, **3**, **3**, **2**, and **1**).

Once an operator obtains `.class` files from a build artifact, container image, file-read bug, or classloader dump, CFR rebuilds readable Java source in seconds. Developers must never store static credentials, signing keys, or unauthenticated administrative URLs inside compiled `.class` bytecode.

Monitoring Media-Processing Outbound Requests (8 Entries)

The media and callback cluster holds **8** sized files totaling **2,418,477** bytes. Four files serve test media: a Python media script (**2,041** B) and a test MP4 container (**6,930** B). Two companion files store the media server log (**1,032,656** B) and output (**410,730** B). Two callback files run and log an HTTP listener (**1,039**-byte script and **399,401**-byte log). The remaining two files run and log a TCP/LDAP listener (**849**-byte script and **564,831**-byte log).

Both listener logs contain lines carrying FFmpeg media-library client strings (`[REF-06]`):

- **HTTP callback log (399,401 bytes; 2,307 lines).** Carries internal timestamps dated `13-14 October 2026` and records **2** `Lavf/58.29.100` **HTTP** `User-Agent` **lines** plus **1** `ffprobe` **line** (client IPs and scanner domains withheld).
- **TCP/LDAP listener log (564,831 bytes; 2,751 lines).** Carries internal timestamps dated `13-14 October 2026` , logging **1,375** TCP connection lines and **1** `Lavf/58.29.100` **line** (client IPs withheld).

`Lavf/58.29.100` is the `libavformat` network client identifier from FFmpeg **4.2.x** (`[REF-06]`). Without `-protocol_whitelist` limits (`[REF-06]` , `[REF-08]`), `ffmpeg` or `ffprobe` workers given external URLs or HLS playlists initiate outbound HTTP or TCP requests. Most of the **2,307** HTTP log lines and **1,375** LDAP log connections reflect routine internet scan noise. Those **3** `Lavf/58.29.100` **lines and 1** `ffprobe` **line** are consistent with media-tool requests to the listeners; they do not identify the client application or confirm SSRF.

4. STOMP-over-WebSocket Brokers, JNDI Lookups, and Staged Shell Files

Alongside the captured heap snapshots and `Lavf/58.29.100` callback logs, the server holds staged WebSocket broker, Java naming, and Linux shell scripts. Figure 4 shows the directory layout across `(root)` and the eight top-level folders.

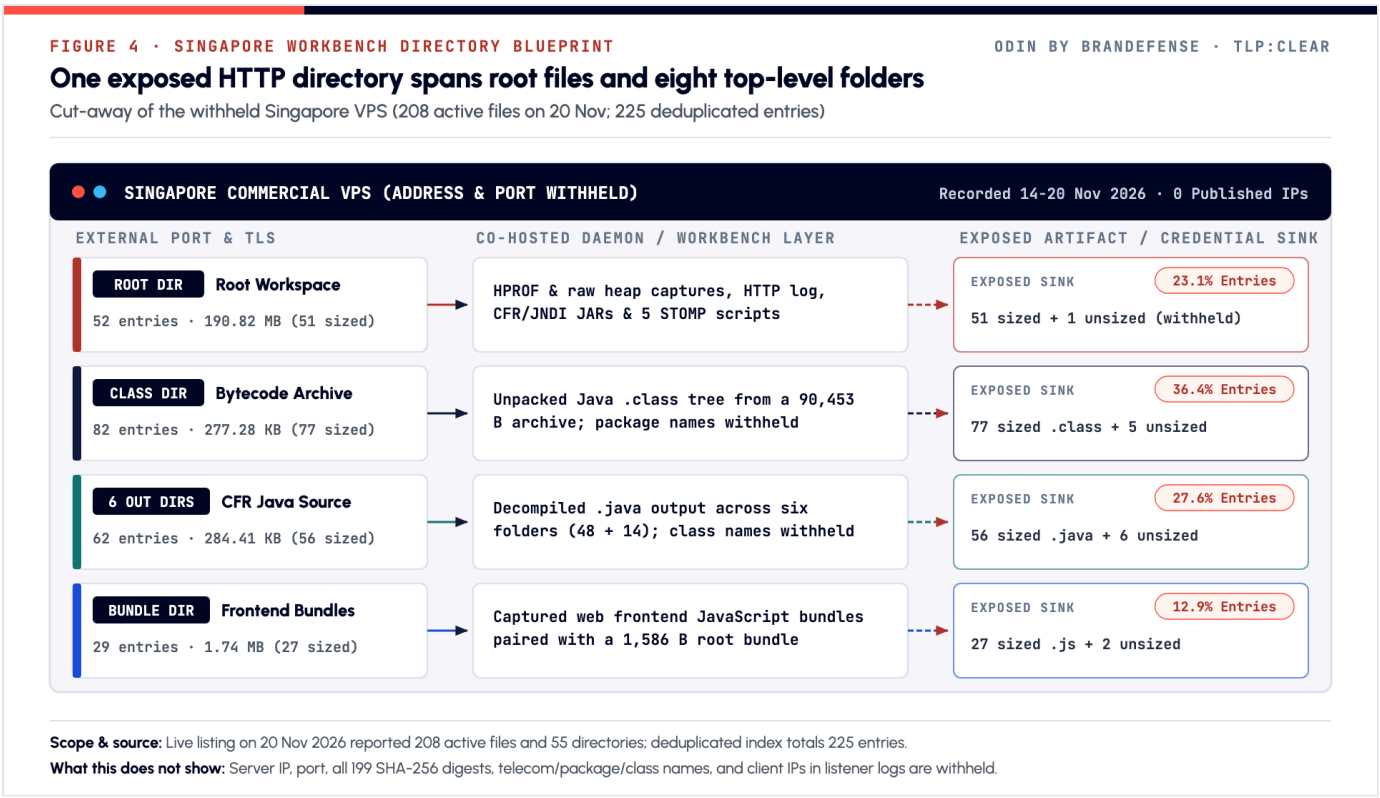

Figure 4: One exposed HTTP directory spans root files and eight top-level folders

Hardening STOMP-over-WebSocket Brokers and Java Naming Lookups (11 Entries)

Eleven sized files (**11,686,790** bytes) cover STOMP messaging scripts and JNDI naming files. Root holds five Python STOMP scripts totaling **27,844** bytes. Their file sizes are **6,467** B, **5,921** B, **5,495** B, **5,011** B, and **4,950** B. Spring WebSocket services often expose STOMP (`[REF-04]` , `[REF-05]`) to route `CONNECT` , `SUBSCRIBE` , and `SEND` frames (`[REF-05]`).

NVD lists Spring Framework `5.0.0` – `5.0.4` and `4.3.0` – `4.3.14` for `CVE-2018-1270` . Spring's advisory also lists `4.3.15` ; both sources describe STOMP `selector` remote code execution (`[REF-01]` , `[REF-02]`). Defenders should upgrade to `5.0.5` or, on the **4.3.x** branch, `4.3.16` ; Spring's current advisory lists `4.3.15` as affected (`[REF-02]`). NVD retains the earlier `4.3.15` boundary (`[REF-01]`).

We cite `CVE-2018-1270` **strictly as architectural context** (`[REF-09]`) because the target's Spring version is unknown. Even on modern Spring versions, missing `ChannelInterceptor` destination checks let authenticated WebSocket users subscribe to foreign queues (`[REF-04]`).

Six JNDI files (**11,658,946** B) sit beside those scripts, led by a **10,357,468**-byte JNDI archive and a **9**-byte archive stub. Four JNDI logs include one **1,296,250**-byte log (**3,608** lines; **5 JNDI reference links**)

and three smaller logs (**2,811** B, **2,367** B, and **41** B). These files show staging only, not target code execution. JDK security updates disable remote JNDI class loading by default (`[REF-07]`); unsafe local object factories can still pose a lookup risk.

Staged Shell, Key, Hash-Cracking, and Hosting Files (20 Entries)

Fifteen staged shell, key, and hash entries (**19,529** sized bytes plus **1** unsized `.pot`) sit in root. They include four `.sh` scripts, one run log, one `.c` file, two withheld SSH key pairs, one withheld **123**-byte hash file, and four cracking logs/potfiles.

Five hosting and proxy files (**16,609,194** B) complete the **225**-entry total. They include a **16,468,015**-byte HTTP directory log, a **768**-byte proxy-verification script, two **113**-byte proxy config files, and one **140,185**-byte withheld `.txt` file. Directory presence alone does not tie generic shell, key, or hash files to the STOMP or FFmpeg tracks.

Figure 2 maps the five operational artifact groups (**220** of **225** entries) to network, host, and file detections.

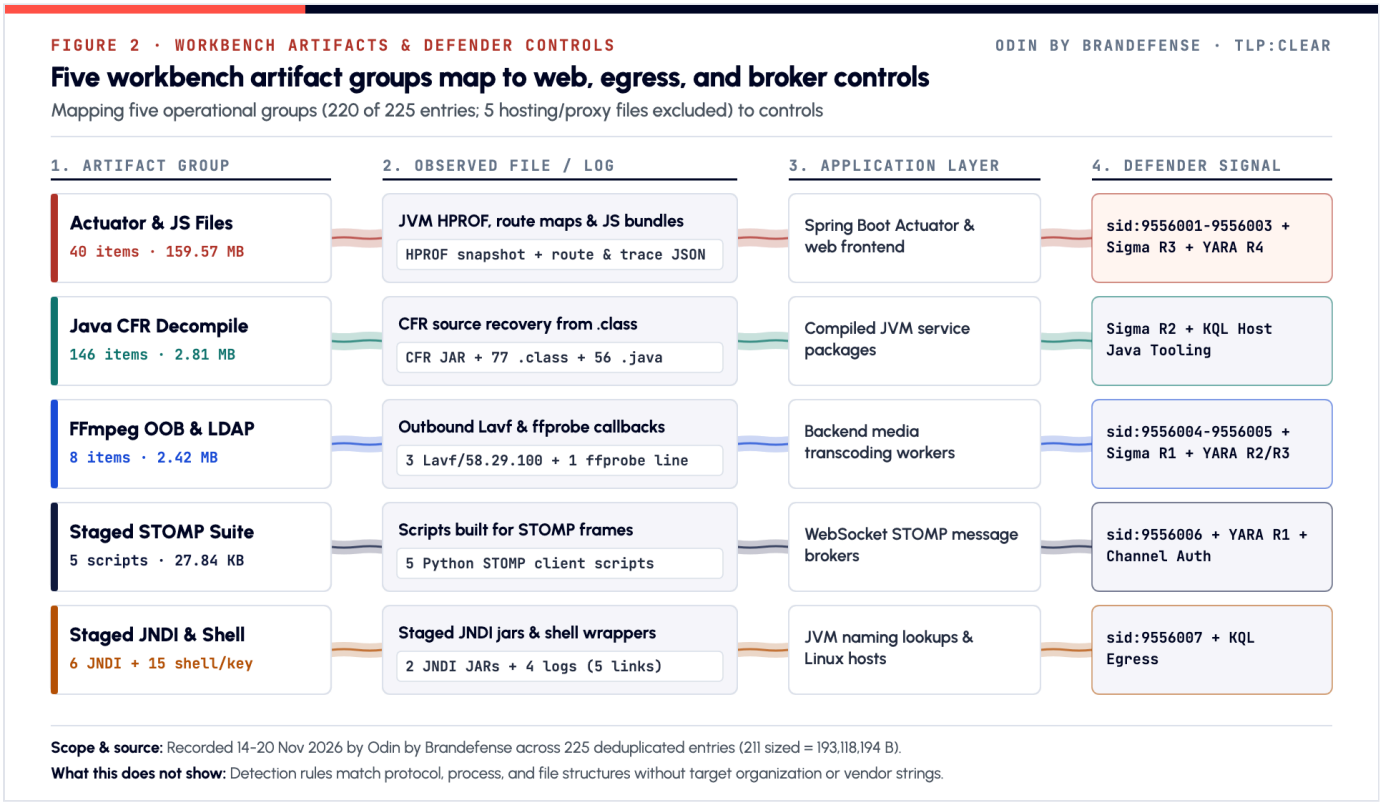

Figure 2: Five workbench artifact groups map to web, egress, and broker controls

5. Enterprise Java, Media, and WebSocket Exposure Analysis

Application and platform teams should verify the four controls below against their own Spring Boot, media, and WebSocket services before treating any misconfiguration as an established root cause:

Exposure layer	Observed workbench evidence	Control to verify	Operational risk
Spring Boot Actuator & frontend bundles	a raw heap capture (117.14 MB), an HPROF snapshot (25.74 MB), extracted heap strings, route and trace captures, and 29 frontend bundle items.	Verify whether `/actuator/*` and `/heapdump` are disabled and blocked at reverse proxies (`[REF-03]`); exact acquisition path is unverified.	JVM heap snapshots expose database credentials, session tokens, signing keys, and route tables (`[REF-10]`).
Decompiled JVM `.class` packages	A compressed bytecode archive (90,453 B), 82 `.class` entries (77 sized), a CFR decompiler archive (2,162,315 B), and 62 `.java` entries (56 sized).	Verify whether compiled `.class` archives embed static secrets or unauthenticated internal endpoints; archive origin is unverified.	CFR decompilation turns leaked bytecode into readable controller annotations and parameter sinks.
Media-worker outbound fetches (`FFmpeg`)	A Python media script, a test MP4 file, an HTTP callback log (2 `Lavf/58.29.100` + 1 `ffprobe`), and a TCP/LDAP log (1 `Lavf/58.29.100`).	Verify whether `ffmpeg` and `ffprobe` enforce `-protocol_whitelist file` inside egress-blocked subnets (`[REF-06]` , `[REF-08]`).	Unvalidated media URLs let backend transcoding workers probe internal services or external listeners.

Exposure layer	Observed workbench evidence	Control to verify	Operational risk
STOMP brokers & JVM naming lookups	5 Python STOMP scripts, two JNDI archives, and four logs (5 JNDI links); intent unverified.	Verify Spring Framework versions (`>= 5.0.5` or `4.3.16` , modern 6.x), STOMP `ChannelInterceptor` checks, and JNDI egress blocks (`[REF-01]` , `[REF-04]` , `[REF-07]`).	Missing topic checks expose cross-user WebSocket queues; legacy Spring or unsafe JNDI sinks risk code execution.

6. Alternative Hypotheses

The table below evaluates four competing hypotheses regarding the workbench's purpose, log activity, and staged scripts.

Hypothesis	Evidence for	Evidence against	Assessment
Staged STOMP and JNDI scripts achieved RCE via `CVE-2018-1270`	Directory holds 5 Python STOMP scripts, two JNDI archives, four JNDI logs (5 JNDI links), shell wrappers, and a 123-byte hash file.	Target Spring version is unknown; the JNDI logs reflect listener setup rather than target class loading, and root key/hash files have unverified provenance.	**Unverified.** Cite `CVE-2018-1270` strictly as architectural context and treat STOMP/JNDI files as staged intent.
All 1,375 TCP/LDAP connections and 2,307 HTTP log lines came from targets	Both listener logs hold thousands of entries starting on `13-14 October 2026` .	Public HTTP and TCP listeners log automated internet scanners and self-tests; only 3 lines match `Lavf/58.29.100` and 1 matches `ffprobe` .	**Refuted for total line counts.** Those four lines are consistent with media-tool requests to the listeners; they do not identify the client application or confirm SSRF.

Hypothesis	Evidence for	Evidence against	Assessment
Authorized penetration test or developer debug host	Uses standard CFR Java decompilation and six structured output folders.	Exposes **142.88 MB** of heap snapshots, target `.class` packages, SSH private keys, and cracking potfiles on an open HTTP directory on a foreign VPS.	**Unresolved.** Public exposure is unsafe but does not establish who owned or used the workspace.
Cohesive application assessment workspace	Root classloader, header, STOMP, and media scripts align with the extracted bytecode folder, frontend bundle items, and heap captures.	Generic shell scripts and hash logs in root could come from lab tests or separate targets.	**Supported for the Java and media tracks**, with explicit boundaries on generic shell and hash files.

7. Validated Detection Rules

The Singapore VPS IPv4 address, host port, and all **199** file SHA-256 digests are withheld. Defenders should deploy **protocol, process, and file-structure rules** that detect this tradecraft on any network. Static checks passed in the Odin Suricata linter (**0** problems) and offline structural checks across KQL, **3** Sigma rules, and **4** `filesize`-bounded YARA rules. Because live `suricata -T`, `Kusto.Language`, `pySigma`, and `YARA` engines were not run locally, teams should test each rule in staging before production rollout.

7.1 Suricata 7.x Network Signatures (`sid:9556001`–9556007)

Three `alert` rules monitor inbound `GET` requests to `/heapdump` (`sid:9556001`) and `/actuator/mappings` (`sid:9556003`), plus outbound `JAVA PROFILE 1.0.` responses (`sid:9556002`). Two rules alert on outbound `Lavf/` and `ffprobe` HTTP `User-Agent` headers (`sid:9556004` and `sid:9556005`). The final two rules cover external `.stomp` WebSocket upgrades (`sid:9556006`) and inbound `${jndi:` HTTP request bodies (`sid:9556007`).

All seven HTTP `alert` rules generate alerts without blocking inline. They require cleartext HTTP traffic or placement behind a TLS-terminating reverse proxy or forward proxy, and `sid:9556006` should be tuned if external browsers legitimately use `.stomp` subprotocols.

```
SURICATA

# Odin by Brandefense - Monograph 56 (Exposed Java Workbench) Suricata 7.x Ruleset
# Static syntax checked with Odin Suricata linter (0 problems); test with suricata -T
before deployment.

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R56 Inbound Spring Boot Actua
tor Heapdump Request"; flow:established,to_server; http.method; content:"GET"; bsize:
3; http.uri; content:"/heapdump"; fast_pattern; classtype:web-application-attack; sid:
9556001; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R56 Outbound JVM HPROF Heapdu
mp Response Body"; flow:established,to_client; http.response_body; content:"JAVA PROFI
LE 1.0."; startswith; fast_pattern; classtype:credential-theft; sid:9556002; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R56 Inbound Spring Boot Actua
tor Mappings Probe"; flow:established,to_server; http.method; content:"GET"; bsize:3;
http.uri; content:"/actuator/mappings"; fast_pattern; classtype:attempted-recon; sid:9
556003; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R56 Outbound FFmpeg Lavf HTTP
User-Agent Callback"; flow:established,to_server; http.user_agent; content:"Lavf/"; st
artswith; fast_pattern; classtype:policy-violation; sid:9556004; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R56 Outbound FFprobe HTTP Use
r-Agent Callback"; flow:established,to_server; http.user_agent; content:"ffprobe"; noc
ase; fast_pattern; classtype:policy-violation; sid:9556005; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R56 Inbound STOMP over WebSoc
ket Upgrade Handshake"; flow:established,to_server; http.header; content:"Upgrade: web
socket"; nocase; content:"Sec-WebSocket-Protocol:"; nocase; content:".stomp"; nocase;
fast_pattern; classtype:protocol-command-decode; sid:9556006; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R56 Inbound JNDI Lookup Expre
ssion in HTTP Body"; flow:established,to_server; http.request_body; content:"${jndi:";
nocase; fast_pattern; classtype:web-application-attack; sid:9556007; rev:1;)
```

7.2 Microsoft Sentinel KQL Hunt

Replace the two placeholder FQDNs in `ApprovedMediaAndBuildHosts` with your approved media-ingestion and Java build hostnames before scheduling.

```
KQL

// Odin by Brandefense - Monograph 56 (Exposed Java Workbench)
// Hunts Spring Boot heapdump/decompiler/JNDI host execution and unexpected FFmpeg or
Java LDAP/RMI public egress.
let Lookback = 7d;
let ApprovedMediaAndBuildHosts = dynamic(["media-worker-01.corp.example", "ci-java-01.
corp.example"]);
let HostJavaAndMediaAnomalies =
```

```
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedMediaAndBuildHosts))
    | where (
        (FileName in~ ("ffmpeg", "ffmpeg.exe", "ffprobe", "ffprobe.exe") and (ProcessC
ommandLine contains "http://" or ProcessCommandLine contains "https://" or ProcessComm
andLine contains "tcp://" or ProcessCommandLine contains "ldap://" or ProcessCommandLi
ne contains "ftp://"))
        or (FileName in~ ("java", "java.exe") and (ProcessCommandLine contains "--outp
utdir" or ProcessCommandLine contains "--extraclasspath" or ProcessCommandLine contain
s "ldap://"))
        or (ProcessCommandLine contains "/actuator/heapdump" or ProcessCommandLine con
tains ".hprof" or ProcessCommandLine contains "jmap -dump:")
    )
    | project Timestamp, DeviceName, AccountName, SignalType = "HostMediaOrJavaToolin
g", Detail = ProcessCommandLine;
let MediaAndJndiUnexpectedEgress =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedMediaAndBuildHosts))
    | where RemoteIPType == "Public"
    | where (
        InitiatingProcessFileName in~ ("ffmpeg", "ffmpeg.exe", "ffprobe", "ffprobe.ex
e")
        or (InitiatingProcessFileName in~ ("java", "java.exe") and RemotePort in (389,
636, 1099, 1389))
    )
    | project Timestamp, DeviceName, AccountName = InitiatingProcessAccountName, Signa
lType = "UnexpectedMediaOrJndiEgress", Detail = strcat(InitiatingProcessFileName, " ->
", RemoteIP, ":", tostring(RemotePort), " (", RemoteUrl, ")");
union HostJavaAndMediaAnomalies, MediaAndJndiUnexpectedEgress
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), EventCount = count
(), Signals = make_set(SignalType, 6), SampleDetails = make_set(Detail, 5) by DeviceNa
me, AccountName
| order by LastSeen desc
```

7.3 Portable Sigma and YARA Rules

- **Sigma (`sigma_r56_ffmpeg_ssrf_cfr_and_jndi.yml`).** Rule 1 detects `/ffmpeg` or `/ffprobe` passing `http`, `https`, `tcp`, `ldap`, or `ftp` URL schemes outside `media-worker-01.corp.example`. Rule 2 catches `/java` passing `--outputdir`, `--extraclasspath`, or `ldap://` outside `ci-java-01.corp.example`. Rule 3 detects `/actuator/heapdump`, `.hprof`, or `jmap -dump:` outside `apm-diag-01.corp.example`.

- **YARA (`odin_r56_stomp_jndi_and_oob_artifacts.yar`).** Rules 1 and 2 detect STOMP-over-WebSocket test scripts and raw-socket Python LDAP or OOB listeners. Rules 3 and 4 detect HTTP callback listener logs containing a `Lavf` identifier and an HTTP request marker, plus Actuator heapdump or classloader parser scripts.

8. Behavioral and Protocol Hunt Markers (0 Published IPs or Workbench SHA-256s)

Zero server IPv4 addresses or workbench-file SHA-256 digests appear in the indicator feeds. The server holds target heap snapshots, compiled `.class` files, and withheld key and hash files. Defenders can hunt using the 10 behavioral markers below.

Marker category	Pattern / protocol artifact	Layer	Defensive hunt & control
Actuator heap route	`GET /actuator/heapdump` (or `/heapdump`)	HTTP / proxy ingress	Block `/actuator/*` and `/heapdump` at reverse proxies and alert via `sid:9556001`.
JVM HPROF magic header	`JAVA PROFILE 1.0.`	HTTP response stream	Alert on `HPROF` headers in decrypted HTTP responses (`sid:9556002`).
Actuator route map	`GET /actuator/mappings` & `/actuator/httptrace`	HTTP / proxy ingress	Disable route-mapping and trace endpoints in production (`[REF-03]`; `sid:9556003`).

Marker category	Pattern / protocol artifact	Layer	Defensive hunt & control
FFmpeg HTTP callback	`User-Agent: Lavf/58.29.100` (`Lavf/`)	HTTP egress	Alert on `Lavf/` outbound requests at web proxies (`sid:9556004`; 3 hits in logs).
FFprobe stream callback	`User-Agent: ffprobe`	HTTP egress	Alert on `ffprobe` outbound HTTP fetches (`sid:9556005`; 1 hit in HTTP log).
STOMP WebSocket upgrade	`Sec-WebSocket-Protocol: v10.stomp` – `v12.stomp`	WebSocket ingress	Audit unexpected external `.stomp` upgrades (`sid:9556006`) and require authentication (`[REF-04]`).
STOMP frame headers	`SUBSCRIBE` / `SEND` / `selector:`	STOMP broker	Enforce Spring Security `ChannelInterceptor` topic checks (`[REF-04]`; YARA Rule 1).
JNDI lookup & egress	`${jndi:` & JVM egress to 389, 636, 1099, 1389	HTTP / TCP egress	Alert on `${jndi:` bodies (`sid:9556007`) and block JVM LDAP/RMI egress (KQL; `[REF-07]`).

Marker category	Pattern / protocol artifact	Layer	Defensive hunt & control
Bytecode decompiler CLI	`java` with `--outputdir` or `--extraclasspath`	Linux / Windows EDR	Alert on decompiler or `ldap://` flags outside build hosts (Sigma Rule 2; KQL).
Media URL CLI fetch	`ffmpeg` / `ffprobe` remote URL schemes	Linux / Windows EDR	Enforce `-protocol_whitelist file` (`[REF-06]`) and alert via Sigma Rule 1 and KQL.

9. Hardening Priorities for Spring Boot, Media, and Messaging Tiers

Application, media, and network defenders should apply five controls in order:

1. **Disable external Actuator routes and rotate heap secrets (`0-48 Hours`).** Exclude `heapdump` , `mappings` , `httptrace` , `httpexchanges` , `env` , and `threaddump` in production (`[REF-03]`). Bind management ports to internal admin interfaces, block `/actuator/*` and `/heapdump` at reverse proxies, and rotate secrets after any heap exposure (`[REF-10]`).
2. **Enforce FFmpeg protocol allowlists and isolate worker egress (`0-72 Hours`).** Pass `-protocol_whitelist file,pipe,crypto` to `ffmpeg` and `ffprobe` so network protocols stay disabled (`[REF-06]`). Read media from local storage, reject external HLS playlist URLs, and block worker subnets from RFC1918, `169.254.169.254` , and internet egress (`[REF-08]`).
3. **Patch Spring messaging and authorize STOMP destinations (`Days 1-7`).** Confirm no services run Spring Framework `5.0.0` – `5.0.4` or `4.3.0` – `4.3.15` (`CVE-2018-1270`). Upgrade to `5.0.5` or, on the **4.3.x** branch, `4.3.16` ; Spring's current advisory lists `4.3.15` as affected (`[REF-02]`). NVD retains the earlier `4.3.15` boundary (`[REF-01]`). Enforce Spring Security `ChannelInterceptor` checks on `CONNECT` , `SUBSCRIBE` , and `SEND` frames (`[REF-04]` , `[REF-05]`).

4. **Lock down JNDI lookups and outbound LDAP/RMI ports (`Days 7-14`).** Run current LTS JDKs where remote JNDI class loading defaults to `false` (`[REF-07]`). Block untrusted strings from `InitialContext.lookup()` and block outbound `TCP 389` , **636**, **1099**, and **1389**.
5. **Audit `.class` packages and `.js` bundles in CI (`Days 14-30`).** Scan Java archives and frontend bundles before release to strip static keys, internal hostnames, and debug route tables.

10. Scope, Limits, and External References

Odin by Brandefense analyzed workbench artifacts recorded between `14 and 20 November 2026` on a Singapore VPS (address and port withheld). On `20 November 2026` , the open directory reported **208** active files and **55** directories. Across `14-20 November 2026` , deduplicated indexing captured **225** entries (**199** unique SHA-256 digests). Those entries span **211** sized files totaling **193,118,194** bytes and **14** unsized entries.

The two listener logs carry internal timestamps dated **13** and `14 October 2026` . Staged STOMP scripts (**5**), JNDI files (**6**), and shell scripts do not establish target execution, and `CVE-2018-1270` is cited strictly as architectural context. The VPS IPv4 address, host port, and all **199** file SHA-256 digests are withheld, alongside all organization, telecom brand, Java package/class, scanner-domain, client-IP, and captured filenames.

1. `[REF-01]` NIST National Vulnerability Database, *CVE-2018-1270 Detail: Spring Framework STOMP over WebSocket Remote Code Execution* (verified on 24 November 2026).
2. `[REF-02]` VMware / Spring Security Advisory, *CVE-2018-1270: Remote Code Execution with spring-messaging* (verified on 24 November 2026).
3. `[REF-03]` Spring Boot Reference Documentation, *Production-Ready Endpoints (Actuator)* (verified on 24 November 2026).
4. `[REF-04]` Spring Framework Reference Documentation, *STOMP over WebSocket* (verified on 24 November 2026).
5. `[REF-05]` STOMP Specification, *Simple Text Oriented Messaging Protocol 1.2* (verified on 24 November 2026).
6. `[REF-06]` FFmpeg Official Documentation, *FFmpeg Protocols Documentation (`protocol_whitelist`)* (verified on 24 November 2026).

7. [REF-07] Dev.java, *JDK Tools and Updates to Help Safeguard Java Applications* (verified on 25 November 2026).
8. [REF-08] OWASP Cheat Sheet Series, *Server-Side Request Forgery Prevention Cheat Sheet* (verified on 24 November 2026).
9. [REF-09] MITRE ATT&CK, *Exploit Public-Facing Application (T1190)* (verified on 24 November 2026).
10. [REF-10] MITRE ATT&CK, *Unsecured Credentials (T1552)* (verified on 24 November 2026).

BRANDEFENSE CAPABILITIES RELATED TO THIS RESEARCH

Cyber Threat Intelligence

Track the actors, infrastructure and campaigns behind threats like these.

Vulnerability Intel

Prioritise the flaws that attackers actually stage.

External Attack Surface

Find exposed middleware before it is targeted.

DEFENDER KIT

Indicators of compromise

Verified adversary infrastructure and tool hashes from this study. Every value below is also discussed in the report.

No network or file indicators 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 below to find the same exposure in your own estate.

10 records from this study are not published in this table: candidate, suspected and dual-use hosts, victim-related values, and hashes of exposed secrets or files.

DEFENDER KIT

Detection rules

The rule files exactly as printed in the report. Test them in your own engine before deployment.

Suricata **odin_r56_actuator_ffmpeg_and_stomp_hunt.rules** 11 lines

```
# Odin by Brandefense - Monograph 56 (Exposed Java Workbench) Suricata 7.x Ruleset
# Static syntax checked with Odin Suricata linter (0 problems); test with suricata -T
before deployment.

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R56 Inbound Spring Boot Actua
tor Heapdump Request"; flow:established,to_server; http.method; content:"GET"; bsize:
3; http.uri; content:"/heapdump"; fast_pattern; classtype:web-application-attack; sid:
9556001; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R56 Outbound JVM HPROF Heapdu
mp Response Body"; flow:established,to_client; http.response_body; content:"JAVA PROFI
LE 1.0."; startswith; fast_pattern; classtype:credential-theft; sid:9556002; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R56 Inbound Spring Boot Actua
tor Mappings Probe"; flow:established,to_server; http.method; content:"GET"; bsize:3;
http.uri; content:"/actuator/mappings"; fast_pattern; classtype:attempted-recon; sid:9
556003; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R56 Outbound FFmpeg Lavf HTTP
User-Agent Callback"; flow:established,to_server; http.user_agent; content:"Lavf/"; st
artswith; fast_pattern; classtype:policy-violation; sid:9556004; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R56 Outbound FFprobe HTTP Use
r-Agent Callback"; flow:established,to_server; http.user_agent; content:"ffprobe"; noc
ase; fast_pattern; classtype:policy-violation; sid:9556005; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R56 Inbound STOMP over WebSoc
ket Upgrade Handshake"; flow:established,to_server; http.header; content:"Upgrade: web
socket"; nocase; content:"Sec-WebSocket-Protocol:"; nocase; content:".stomp"; nocase;
fast_pattern; classtype:protocol-command-decode; sid:9556006; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R56 Inbound JNDI Lookup Expre
ssion in HTTP Body"; flow:established,to_server; http.request_body; content:"${jndi:";
nocase; fast_pattern; classtype:web-application-attack; sid:9556007; rev:1;)
```

Show full rule

YARA **odin_r56_stomp_jndi_and_oob_artifacts.yar** 76 lines

```
/*
   Odin by Brandefense - Monograph 56 (Exposed Java Workbench) YARA Suite
   4 filesize-bounded structural rules covering STOMP-over-WebSocket test scripts,
   custom LDAP/OOB callback listeners, FFmpeg Lavf HTTP callback logs, and Actuator
```

```
   heapdump/classloader parsers (zero victim, telecom, or vendor identifiers).
*/

rule ODIN_R56_STOMP_WebSocket_Probe_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "56"
        description = "Detects Python scripts crafting raw STOMP CONNECT, SUBSCRIBE, a
nd SEND frames over WebSockets"
        date = "2026-11-24"
    strings:
        $f1 = "CONNECT" ascii
        $f2 = "SUBSCRIBE" ascii
        $f3 = "SEND" ascii
        $w1 = "websocket" ascii nocase
        $w2 = "accept-version:1." ascii
        $w3 = "selector:" ascii
    condition:
        filesize < 128KB and $f1 and $f2 and $f3 and 2 of ($w*)
}

rule ODIN_R56_Custom_LDAP_OOB_Callback_Listener
{
    meta:
        author = "Odin by Brandefense"
        monograph = "56"
        description = "Detects lightweight Python raw-socket LDAP or HTTP out-of-band
callback listeners"
        date = "2026-11-24"
    strings:
        $s1 = "socket.socket" ascii
        $s2 = "0.0.0.0" ascii
        $p1 = "1389" ascii
        $p2 = "389" ascii
        $p3 = "Lavf/" ascii
        $p4 = "User-Agent" ascii
    condition:
        filesize < 64KB and $s1 and $s2 and 2 of ($p*)
}

rule ODIN_R56_FFmpeg_Lavf_Callback_Log
{
    meta:
        author = "Odin by Brandefense"
        monograph = "56"
        description = "Detects HTTP callback listener logs containing a Lavf identifie
```

```
r and an HTTP request marker."
        date = "2026-11-24"
    strings:
        $l1 = "Lavf/58." ascii
        $l2 = "User-Agent:" ascii nocase
        $l3 = "ffprobe" ascii nocase
        $h1 = "GET /" ascii
    condition:
        filesize < 5MB and $l1 and $h1 and ($l2 or $l3)
}

rule ODIN_R56_Actuator_Heapdump_and_Classloader_Parser
{
    meta:
        author = "Odin by Brandefense"
        monograph = "56"
        description = "Detects scripts parsing JVM HPROF heapdumps, JMX MBeans, and Sp
ring Boot Actuator route mappings"
        date = "2026-11-24"
    strings:
        $h1 = "heapdump" ascii nocase
        $h2 = "JAVA PROFILE 1.0" ascii
        $m1 = "ClassLoader" ascii
        $m2 = "javax.management" ascii
        $m3 = "/actuator/" ascii
    condition:
        filesize < 128KB and ($h1 or $h2) and 2 of ($m*)
}
```

Sentinel KQL **sentinel_r56_ffmpeg_egress_actuator_and_jndi_hunt.kql** 28 lines

```
// Odin by Brandefense - Monograph 56 (Exposed Java Workbench)
// Hunts Spring Boot heapdump/decompiler/JNDI host execution and unexpected FFmpeg or
Java LDAP/RMI public egress.
let Lookback = 7d;
let ApprovedMediaAndBuildHosts = dynamic(["media-worker-01.corp.example", "ci-java-01.
corp.example"]);
let HostJavaAndMediaAnomalies =
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedMediaAndBuildHosts))
    | where (
        (FileName in~ ("ffmpeg", "ffmpeg.exe", "ffprobe", "ffprobe.exe") and (ProcessC
ommandLine contains "http://" or ProcessCommandLine contains "https://" or ProcessComm
andLine contains "tcp://" or ProcessCommandLine contains "ldap://" or ProcessCommandLi
ne contains "ftp://"))
```

```
        or (FileName in~ ("java", "java.exe") and (ProcessCommandLine contains "--outp
utdir" or ProcessCommandLine contains "--extraclasspath" or ProcessCommandLine contain
s "ldap://"))
        or (ProcessCommandLine contains "/actuator/heapdump" or ProcessCommandLine con
tains ".hprof" or ProcessCommandLine contains "jmap -dump:")
    )
    | project Timestamp, DeviceName, AccountName, SignalType = "HostMediaOrJavaToolin
g", Detail = ProcessCommandLine;
let MediaAndJndiUnexpectedEgress =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedMediaAndBuildHosts))
    | where RemoteIPType == "Public"
    | where (
        InitiatingProcessFileName in~ ("ffmpeg", "ffmpeg.exe", "ffprobe", "ffprobe.ex
e")
        or (InitiatingProcessFileName in~ ("java", "java.exe") and RemotePort in (389,
636, 1099, 1389))
    )
    | project Timestamp, DeviceName, AccountName = InitiatingProcessAccountName, Signa
lType = "UnexpectedMediaOrJndiEgress", Detail = strcat(InitiatingProcessFileName, " ->
", RemoteIP, ":", tostring(RemotePort), " (", RemoteUrl, ")");
union HostJavaAndMediaAnomalies, MediaAndJndiUnexpectedEgress
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), EventCount = count
(), Signals = make_set(SignalType, 6), SampleDetails = make_set(Detail, 5) by DeviceNa
me, AccountName
| order by LastSeen desc
```

Sigma `sigma_r56_ffmpeg_ssrf_cfr_and_jndi.yml` 86 lines

```
title: Exposed Java Workbench FFmpeg or FFprobe Remote URL Network Fetch
id: 8b41c9e2-3d18-4a72-9f05-56a10d92e301
status: experimental
description: Detects FFmpeg or FFprobe processes invoked with remote network protocol
schemes on servers outside approved media-ingestion worker hosts.
author: Odin by Brandefense
date: 2026-11-24
logsource:
  category: process_creation
  product: linux
detection:
  selection_bin:
    Image|endswith:
      - '/ffmpeg'
      - '/ffprobe'
  selection_net:
```

```
    CommandLine|contains:
      - 'http://'
      - 'https://'
      - 'tcp://'
      - 'ldap://'
      - 'ftp://'
  filter_media_workers:
    ComputerName:
      - 'media-worker-01.corp.example'
  condition: selection_bin and selection_net and not filter_media_workers
falsepositives:
  - Approved remote stream ingestion workers fetching allowlisted upstream media origi
ns
level: high
tags:
  - attack.initial-access
  - attack.t1190
---
title: Exposed Java Workbench Bytecode Decompiler or JNDI CLI Invocation
id: 8b41c9e2-3d18-4a72-9f05-56a10d92e302
status: experimental
description: Detects Java command lines passing bytecode decompiler output-directory f
lags or LDAP naming URLs on non-build hosts.
author: Odin by Brandefense
date: 2026-11-24
logsource:
  category: process_creation
  product: linux
detection:
  selection_java:
    Image|endswith: '/java'
  selection_args:
    CommandLine|contains:
      - '--outputdir'
      - '--extraclasspath'
      - 'ldap://'
  filter_build_hosts:
    ComputerName:
      - 'ci-java-01.corp.example'
  condition: selection_java and selection_args and not filter_build_hosts
falsepositives:
  - Authorized software audit or reverse-engineering workstations
level: high
tags:
  - attack.credential-access
  - attack.t1552
---
```

```
title: Exposed Java Workbench JVM Heapdump Acquisition or String Extraction
id: 8b41c9e2-3d18-4a72-9f05-56a10d92e303
status: experimental
description: Detects command lines requesting Spring Boot Actuator heapdump endpoints,
writing HPROF heap snapshots, or running jmap heap dumps outside approved diagnostic h
osts.
author: Odin by Brandefense
date: 2026-11-24
logsource:
  category: process_creation
  product: linux
detection:
  selection_heap:
    CommandLine|contains:
      - '/actuator/heapdump'
      - '.hprof'
      - 'jmap -dump:'
  filter_apm_hosts:
    ComputerName:
      - 'apm-diag-01.corp.example'
  condition: selection_heap and not filter_apm_hosts
falsepositives:
  - Scheduled JVM memory troubleshooting performed by SRE teams on designated diagnost
ic nodes
level: high
tags:
  - attack.credential-access
  - attack.t1552
```

