FULL REPORT · OPERATIONS · MONOGRAPH 58 / 57 · 16 MIN READ

Operation FalsePort: Rogue MySQL JDBC File Reads and Driver Audits

Inside a 1,039-entry operator workbench pairing MySQL, PostgreSQL, and SQL Server JDBC driver audits with live TCP 3306 rogue MySQL listeners and seven captured target files

BY
Odin by Brandefense

PUBLISHED
19 November 2026

READING TIME
16 min read

EDITION
Version 1.1 · TLP:CLEAR

SCOPE
Prepared November 2026

1. Executive Summary

IN SHORT

Operation FalsePort pairs JDBC driver source audits across MySQL, PostgreSQL, and SQL Server with rogue TCP 3306 MySQL listeners on a China-hosted server. Odin by Brandefense analyzed 1,039 deduplicated entries recorded on 13–20 November 2026. Stored logs record 2,167 inbound connects, one Connector/J 8.0.28 session, and seven captured files. Patch JDBC drivers, disable local file reads, and block outbound TCP 3306.

Java web portals, BI tools, and schedulers often let admins test database connections from a browser form. When an app passes a raw JDBC string to its driver, the database client turns into an attack surface.

Instead of breaking into a backend SQL host, an attacker points the Java app at an external fake database port. Operation FalsePort pairs multi-engine JDBC code review with live rogue MySQL listeners and seven captured target files.

From 13 to 20 November 2026, Odin by Brandefense analyzed an exposed workbench on a China-hosted cloud server on TCP port 80. On 20 November 2026, that open folder reported 485 active files and 32 subfolders.

Across the 13–20 November 2026 snapshots, the deduplicated dataset holds 1,039 entries. Those entries span 263 sized files totaling 927.36 megabytes and 776 unsized entries across 267 unique SHA-256 digests.

The server stores captured Linux account files, Git credentials, SSH private keys, and a Spring Boot log. Odin by Brandefense publishes zero server IPv4 addresses and zero workbench SHA-256 digests. Both server addresses, all client IPs, folder names, and target filenames are withheld.

1 **Source and bytecode audits across three SQL engines make up 89.22 percent of entries.** Two folders dominate the workbench file count. An audit folder holds 539 entries, including PostgreSQL JDBC source code, while a MySQL folder holds 388 Connector/J source files.

In the root workspace, compiled PostgreSQL and Microsoft SQL Server JDBC JARs sit beside an extracted SQL Server connection class. Root also stages Apache Shiro JARs and a 42.67-megabyte JNDI exploit JAR.

2 **Two TCP 3306 listener logs record 2,167 inbound connects and one Connector/J 8.0.28 session.** Log `fake3306c2b.log` holds 3,171 lines and 1,145 inbound connect events. A second log (`fakemysql_test.log`) holds 2,083 lines and 1,022 connect events.

Most raw TCP 3306 connects on a public cloud IP come from internet port scanners. Still, `fake3306c2b.log` logs one MySQL Connector/J 8.0.28 client receiving 3,432-byte and 1,686-byte payloads.

3 **A seven-file captured-output folder holds 21.63 megabytes of returned Linux and Spring Boot data.** Inside a seven-file captured-output folder (name withheld), 30 entries include seven sized files totaling 21,627,228 bytes. Those files include a 941-byte `/etc/passwd` file, Git credentials, two SSH private keys, configs, and a 21.59-megabyte Spring Boot log.

4 **An unlinked India-hosted VPS on TCP 8001 shares zero non-public file hashes.** On 20 November 2026, a second open workbench on an India-hosted VPS reported 713 active files (874 entries; 506 unique digests). Comparing all 267 primary digests with all 506 digests on that second host shows zero shared non-public hashes, and nothing links the two operators.

Three Decisions for Platform and Security Leaders

1. **Lock down JDBC DataSource URLs and strip risky client flags.** Build database URLs from checked host, port, and schema fields only. Reject local file-read, deserialize, query-interceptor, and socket-factory options at the API layer.
2. **Upgrade MySQL Connector/J and PostgreSQL pgjdbc drivers.** Update MySQL Connector/J to version 8.2.0 or later and update PostgreSQL pgjdbc to version 42.2.25 or 42.3.2 or later. Check bundled drivers inside BI tools and app containers.
3. **Block outbound SQL ports from Java app subnets.** Stop web and middleware hosts from opening outbound TCP 3306, 5432, 1433, or 1389 connections to public IPs. Run Java apps under accounts without SSH keys or Git credential files.

2. Workbench Architecture, 1,039-Entry Dataset, and Parallel Comparison

Operation FalsePort ran from a China-hosted cloud server on **TCP 80** (address withheld), recorded on `13-20 November 2026`. On `20 November 2026`, the open folder reported **485 active files** and **32 subfolders**.

Across the `13-20 November 2026` snapshots, the deduplicated dataset holds **1,039 entries**. It spans **263** sized files totaling **927,357,421 bytes (927.36 MB)** and **776** unsized entries across **267 SHA-256 digests**. Platform teams can use external attack surface management to find exposed Java panels before attackers change DataSource settings. Figure 1 maps the four workbench tracks.

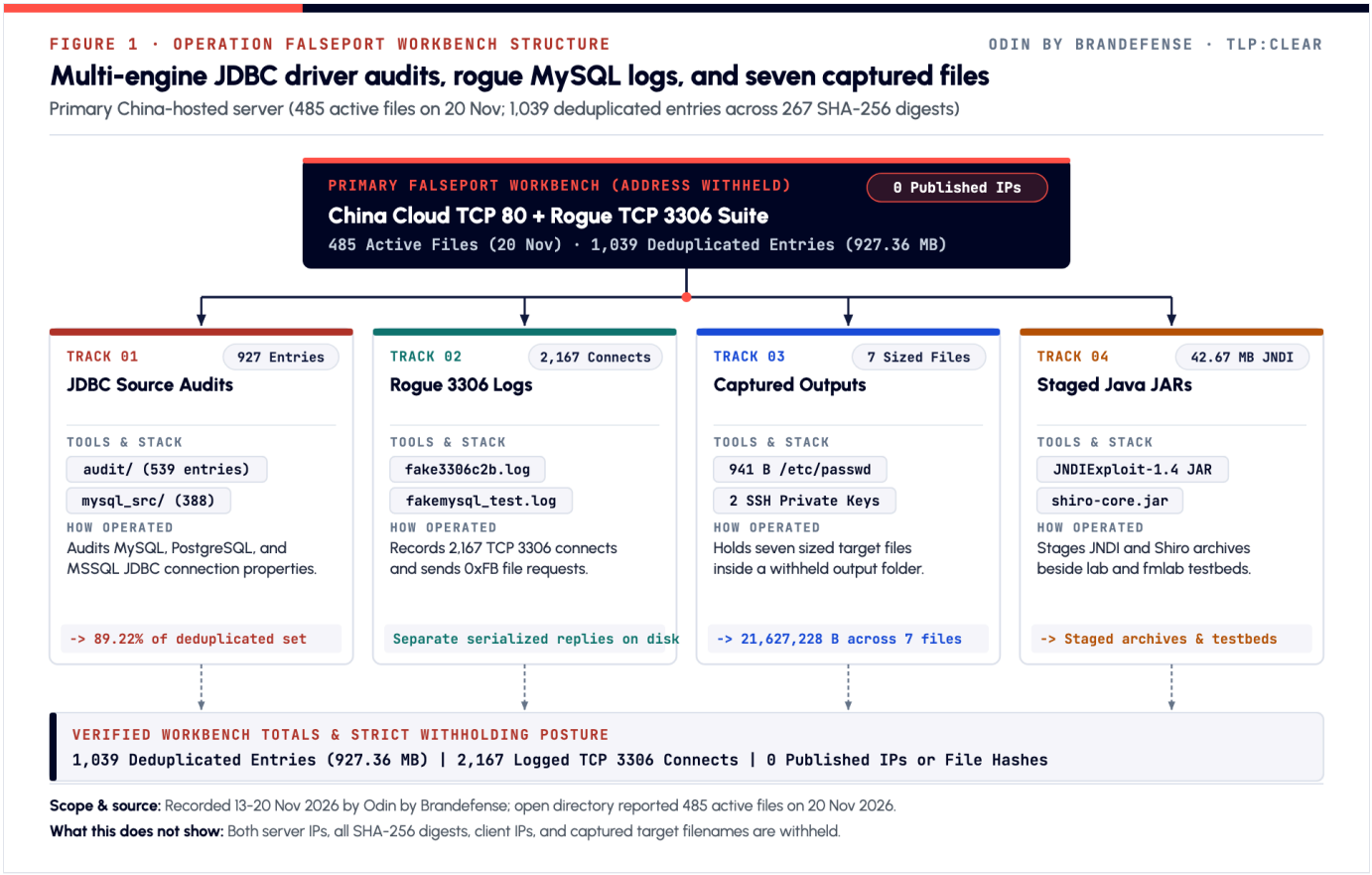

Figure 1: Multi-engine JDBC driver audits, rogue MySQL logs, and seven captured files

Seven-Zone Partition and File-Type Breakdown (`n = 1,039`)

Every table in this report sums to the exact **1,039 entries**, **263 sized files**, **776 unsized entries**, and **927,357,421 sized bytes** on the primary host. Seven top-level zones divide the workspace (`539 + 388 + 52 + 30 + 15 + 14 + 1 = 1,039`).

Top-level workspace zone	Zone count	Entries	Share (%)	Sized files	Unsized entries	Sized bytes	Role on workbench
Multi-Engine Audit Tree (`audit/`)	1	539	51.88%	80	459	583,901	PostgreSQL `pgjdbc` source (`audit/src_pg/`) and cross-driver property checks.
MySQL Driver Source (`mysql_src/`)	1	388	37.34%	114	274	541,993	Extracted `com.mysql.cj.*` Connector/J files for file-read and interceptor checks.
Root Operator Workspace	1 (root)	52	5.00%	40	12	904,597,314	Rogue 3306 logs (`fake3306c2b.log`, `fakemysql_test.log`), `JNDIExploit-1.4-SNAPSHOT.jar`, `pg.jar`, `sqljdbc4.jar`, Shiro JARs, and payloads.
Captured-Output Folder (name withheld)	1	30	2.89%	7	23	21,627,228	Seven sized captures: 941 B `/etc/passwd`, 2 SSH keys, Git creds, configs, and 21.59 MB Spring Boot log.
Driver Test Lab (`lab/`)	1	15	1.44%	11	4	5,319	Local JDBC URL and property test files.
Fake-MySQL Test Lab (`fmlab/`)	1	14	1.35%	11	3	1,666	Rogue MySQL greeting and `0xFB` `LOCAL INFILE` packet tests.
Auxiliary Tools (`tools/`)	1	1	0.10%	0	1	0	Unsized helper folder entry.
Total primary dataset	**6 dirs + root**	**1,039**	**100.00%**	**263**	**776**	**927,357,421**	**485 active files & 32 dirs on `20 Nov 2026`; 267 SHA-256s (withheld).**

Two volume patterns stand out in this layout. Folders `audit/` (**539**) and `mysql_src/` (**388**) hold **89.22% (927 / 1,039)** of all entries while taking only **1.13 MB** of sized text. Meanwhile, the **52** root entries hold **97.55% (904.60 MB)** of total byte volume because root stores large JAR bundles and logs.

The next table splits the same **1,039** entries across five file classes (`684 + 346 + 6 + 2 + 1 = 1,039`). Java source files (**587** `.java`) and JAR archives (**304** `.jar`) make up **85.76%** of all items.

Artifact class	Entries	Share (%)	Sized files	Unsized entries	Sized bytes	Exact extension counts & contents
Text, source & logs (`text`)	684	65.83 %	236	448	881,377,599	587 `.java` , 26 `.log` , 25 `.yml` , 24 `.txt` , 9 `.properties` , 6 `.bin` , 5 `.xml` , 1 `.html` , 1 `.json` .
Archives & bytecode (`exe`)	346	33.30 %	20	326	45,964,797	304 `.jar` (`JNDIExploit-1.4-SNAPSHOT.jar` , `pg.jar` , `sqljdbc4.jar` , `shiro-core.jar` , `shiro-web.jar`), 23 `.bin` , 19 `.class` .
Scripts (`script`)	6	0.58%	5	1	11,664	6 `.py` rogue MySQL listener and JDBC test scripts.
Serialized & other (`other`)	2	0.19%	1	1	1,686	1 `.ser` serialized Java payload (1,686 B) and 1 `.types` file.
Secrets & keys (`secret`)	1	0.10%	1	0	1,675	1 `.bin` captured SSH private key (1,675 B; target names withheld).
Total	**1,039**	**100.00 %**	**263**	**776**	**927,357,421**	**Exact byte parity** (927.36 MB).

Figure 5 combines `lab/` (**15**), `fmlab/` (**14**), and `tools/` (**1**) into one **30**-entry testbed track (**22** sized files = **6,985** B). All five tracks sum to **1,039** entries (`539 + 388 + 52 + 30 + 30 = 1,039`).

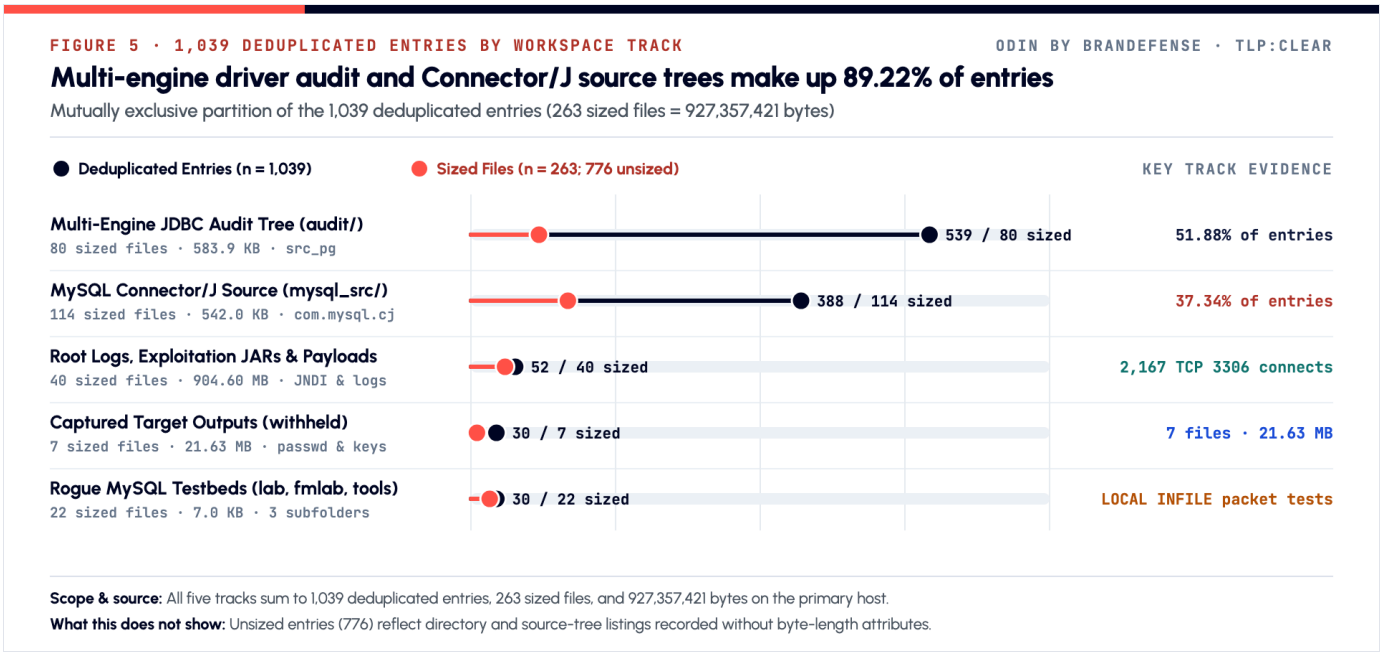

Figure 5: Multi-engine driver audit and Connector/J source trees make up 89.22% of entries

Independent Parallel Workbench: India-Hosted VPS on TCP 8001

On `20 November 2026`, Odin by Brandefense also examined a second open folder on an **India-hosted VPS on TCP 8001** (address withheld). That host reported **713 active files** and **114 folders** on `20 November`. Its deduplicated dataset holds **874 entries**: **715** sized files (**1,577,851,064** bytes, or **1.58 GB**) and **159** unsized entries across **506 SHA-256 digests**.

Four top-level folders divide those **874** entries. Folder `java-chains/` holds **712** entries (versions `2.0.0` / `2.0.1`, **1,311** log lines). Root holds **115** entries, including Linux `/proc/self/environ`, `/proc/1/cmdline`, and `/proc/self/cmdline` captures.

Folder `payloads/` holds **24** entries, while `MySQL_Fake_Server/` holds **23** entries. Comparing the **267** primary SHA-256 digests with the **506** India VPS digests shows **zero shared non-public hashes**, and nothing links the two operators.

3. Multi-Engine JDBC Driver Auditing and Rogue TCP 3306 Listeners

BI dashboards, data tools, and Java admin panels often expose "Test Connection" forms (`[REF-09]`). Defenders using cyber threat intelligence can track how attackers point those forms at external database ports. Figure 3 shows an analytical model of staged URL edits, logged TCP **3306** handshakes, and stored file captures.

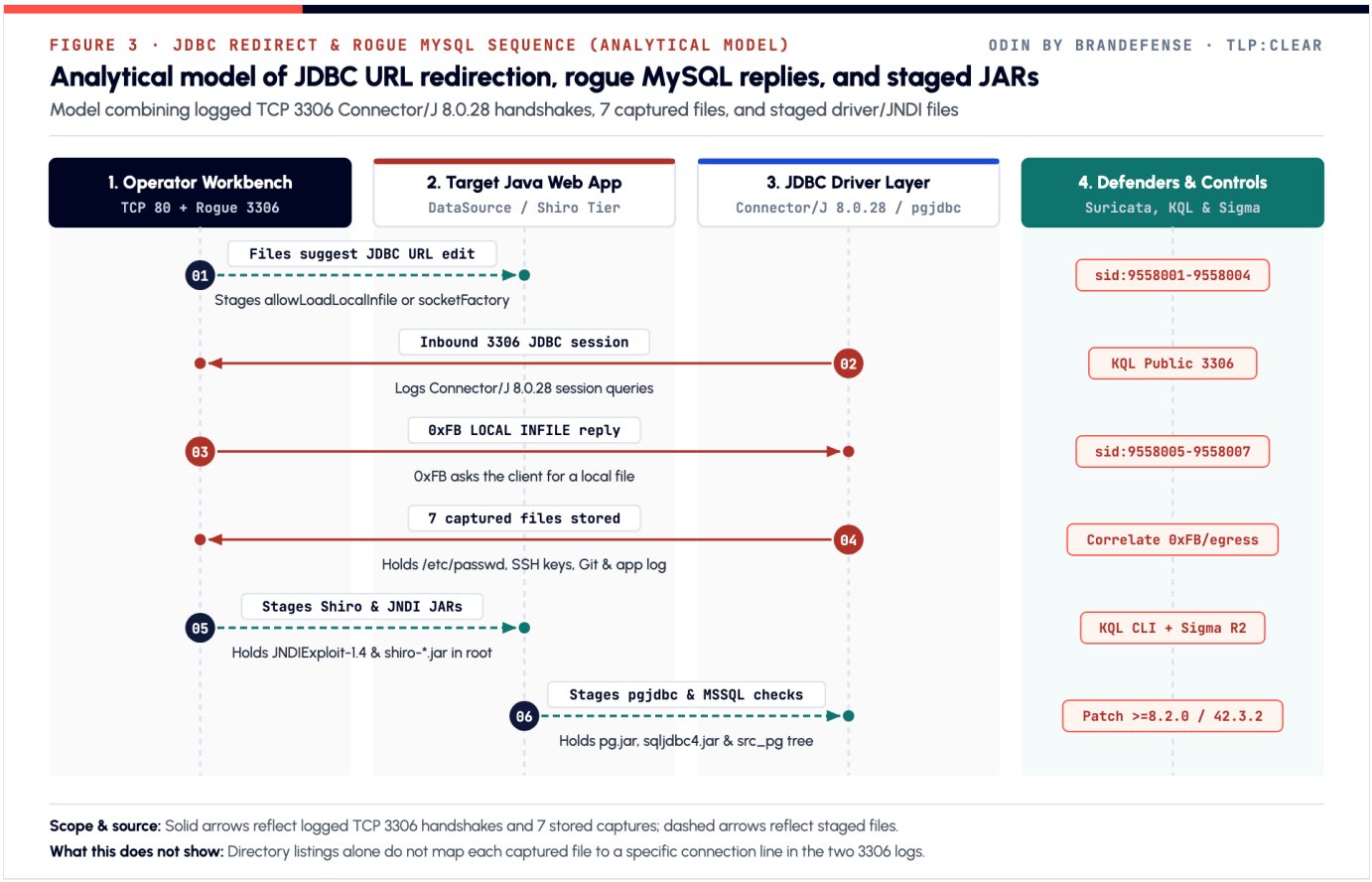

Figure 3: Analytical model of JDBC URL redirection, rogue MySQL replies, and staged JARs

Auditing MySQL, PostgreSQL, and SQL Server JDBC Drivers (927 Entries)

The primary operator unpacked **927** source entries across `mysql_src/` (**388** entries, **541,993** B) and `audit/` (**539** entries, **583,901** B). Root also holds compiled driver JARs and classes:

- **MySQL Connector/J (`com.mysql.cj.*`, `[REF-01]`–`[REF-04]`).** Folder `mysql_src/` holds **388** source files tracing Connector/J 8.x URL flags (`[REF-03]`). Flag `allowLoadLocalInfile=true` lets a server read local JVM files (`[REF-04]`). Flag `autoDeserialize=true` paired with `queryInterceptors=` triggers Java deserialization of server replies. Oracle fixed Connector/J flaws in `CVE-2021-2471` (`8.0.26` and prior; fixed in `8.0.27`, `[REF-02]`) and `CVE-2023-22102` (`8.1.0` and prior; fixed in `8.2.0`, `[REF-01]`).
- **PostgreSQL `pgjdbc` (`audit/src_pg/` and `pg.jar` 1,006,732 B, `[REF-05]`, `[REF-06]`).** Flaw `CVE-2022-21724` affects `pgjdbc` `9.4.1208` before `42.2.25` and `42.3.0` before `42.3.2` (fixed in `42.2.25` and `42.3.2`). Vulnerable builds load classes named in `socketFactory`, `sslfactory`, `sslhostnameverifier`, `sslpasswordcallback`, or `authenticationPluginClassName` before checking interfaces (`[REF-05]`, `[REF-06]`).
- **Microsoft SQL Server JDBC (`sqljdbc4.jar` 584,207 B and `SQLServerConnection.class` 57,341 B, `[REF-07]`).** Root stages `sqljdbc4.jar` and `SQLServerConnection.class` to inspect URL parsing and TLS trust flags (`[REF-07]`). Because no SQL Server or PostgreSQL listener logs appear on disk, both stay staged research.

Live TCP 3306 Rogue MySQL Listener Logs (2,167 Connect Events)

Two root logs record live inbound traffic hitting rogue MySQL listeners on TCP **3306**:

Stored listener log	Sized bytes	Total lines	`== connect from` events	Observed protocol details
`fake3306c2b.log`	151,645	3,171	1,145	Logs inbound TCP 3306 connects (client IPs withheld) and 1 `mysql-connector-java-8.0.28` client running `SELECT @@session.auto_increment_increment...` and `SHOW SESSION STATUS`.
`fakemysql_test.log`	95,593	2,083	1,022	Logs 1,022 inbound TCP 3306 connects during listener tests and exposure (client IPs withheld).
Combined TCP 3306 logs	**247,238**	**5,254**	**2,167**	**Paired on disk with 3,432-byte and 1,686-byte reply payloads.**

Three protocol facts govern those two logs:

1. **Scan noise vs real JDBC clients.** Public TCP **3306** ports draw constant bot scans, so **2,167** connects do not equal **2,167** targets. Real JDBC clients send `mysql-connector-java-8.0.28` strings and `SELECT @@session.auto_increment_increment...` queries. Build `8.0.28` is newer than `8.0.26` (`CVE-2021-2471`) yet falls under `8.1.0` and prior (`CVE-2023-22102`). It also honors `0xFB` file reads when `allowLoadLocalInfile=true` is set (`[REF-03]`).
2. **Server `0xFB` `LOCAL INFILE` requests (`[REF-04]`).** In reply to any SQL query, a fake MySQL server can send byte `0xFB` plus a file path like `/etc/passwd`. When local infile reads are active, the client JVM reads

that local file and streams its bytes back (`[REF-10]`).

3. **Serialized payloads.** Root also holds **3,432**-byte and **1,686**-byte payloads (including the **1,686**-byte `.ser` file) built for `SHOW SESSION STATUS` deserialization tests.

4. Returned Target File Captures and Staged Java Tooling

Operation FalsePort moves beyond code review inside a seven-file captured-output directory (folder name withheld). Figure 4 compares the primary server stack with the unlinked India-hosted VPS.



Figure 4: Primary China-hosted workbench and unlinked India-hosted parallel workbench

Seven Sized Target File Captures (21,627,228 Bytes)

The captured-output folder holds **30 entries**: **23** unsized entries and **7 sized files totaling 21,627,228 bytes (21.63 MB)**. Every filename, user name, and project label is withheld (`21,586,294 + 24,348 + 7,810 + 3,904 + 2,256 + 1,675 + 941 = 21,627,228` bytes):

Capture item	Exact bytes	Share (%)	Generic capture description (names withheld)	Security impact (`[REF-10]`)
Capture 01	21,586,294	99.81%	**Spring Boot app log** (21.59 MB; service name withheld).	Exposes stack traces, internal URLs, tokens, and app state.

Capture item	Exact bytes	Share (%)	Generic capture description (names withheld)	Security impact (`[REF-10]`)
Capture 02	24,348	0.11%	App config or credential file (**24.35 KB**).	Reveals backend service and database settings.
Capture 03	7,810	0.04%	App config or credential file (**7.81 KB**).	Reveals service endpoints or auth state.
Capture 04	3,904	0.02%	**SSH private key** or **Git credentials** (**3.90 KB**).	Supports SSH login or private repo cloning.
Capture 05	2,256	0.01%	**Git credentials** or second **SSH key** (**2.26 KB**).	Pairs with home-folder credential reads from the JVM user.
Capture 06	1,675	0.01%	**SSH private key** (1,675 B, `secret` class in file table).	Matches PEM or OpenSSH private-key length.
Capture 07	941	<0.01 %	Linux `/etc/passwd` file (**941** B; accounts withheld).	Lists OS users and home paths for follow-up `0xFB` reads.
Total	**21,627,228**	**100.00%**	**7 sized files + 23 unsized entries = 30 entries.**	**All target filenames and folder names withheld.**

Reading `/etc/passwd` (**941** B) first lets an operator find the JVM user's home path. Follow-up `0xFB` packets can then request `.ssh/id_rsa`, `.git-credentials`, and Spring Boot logs (**21.59 MB**).

Folder listings alone do not show whether all seven files came from one host or several. They also do not tie each file to a single line in `fake3306c2b.log` or `fakemysql_test.log`.

Staged Apache Shiro and `JNDIExploit-1.4-SNAPSHOT.jar` (42.67 MB)

Root also stages `shiro-core.jar`, `shiro-web.jar`, and `JNDIExploit-1.4-SNAPSHOT.jar` **(42,671,630 bytes)**. Attackers often test Apache Shiro for `rememberMe` or path-filter bypasses on Java admin panels (`[REF-08]`).

Archive `JNDIExploit` stages rogue LDAP and HTTP controllers. Because no Shiro or LDAP callback logs appear in sized files, both JAR sets stay staged capabilities. Figure 2 maps all five workbench groups to Chapter 7 rules.

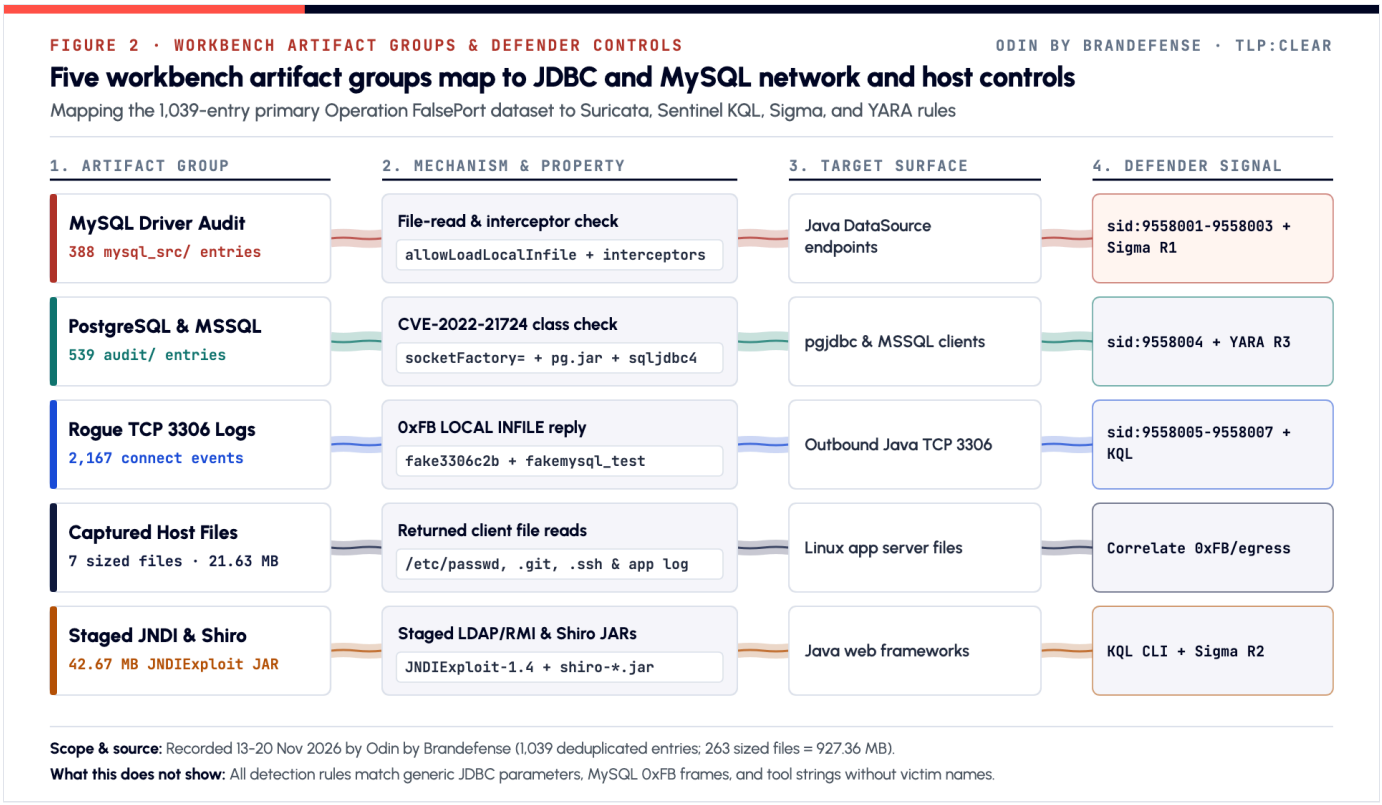

Figure 2: Five workbench artifact groups map to JDBC and MySQL network and host controls

5. Java DataSource and JDBC Exposure Analysis

Platform teams should check four controls across Java web apps and BI tools before treating any gap as a root cause:

Exposure layer	Observed FalsePort evidence	Control to verify	Risk if unguarded
User-editable JDBC URLs	388 `mysql_src/` + 539 `audit/` source entries.	Verify whether admin forms or APIs accept raw `jdbc:` strings or custom flags (`[REF-09]`).	Users can inject `allowLoadLocalInfile=true` , `autoDeserialize=true` , or `socketFactory=` .

Exposure layer	Observed FalsePort evidence	Control to verify	Risk if unguarded
Outdated JDBC drivers	Logged `mysql-connector-java-8.0.28`; staged `pg.jar` & `src_pg/`.	Verify whether WAR/JAR builds ship Connector/J `<= 8.1.0` (`CVE-2023-22102`) or `pgjdbc` `< 42.2.25` / `42.3.2` (`CVE-2022-21724`).	Old drivers allow client driver abuse or unchecked class loading (`[REF-01]`, `[REF-05]`).
Java TCP 3306 egress	2,167 `== connect from` events across 2 TCP 3306 logs.	Verify whether Java subnets can open outbound TCP 3306, 5432, or 1389 sessions to public IPs.	Edited DataSource tests reach external fake MySQL or LDAP ports.
JVM file permissions	7 captures (21.63 MB): `/etc/passwd`, 2 SSH keys, Git creds, log.	Verify whether JVM accounts can read `.ssh/`, `.git-credentials`, and verbose logs (`[REF-10]`).	One `0xFB` exchange returns SSH keys, repo tokens, and app logs.

6. Alternative Hypotheses

Hypothesis	Evidence for	Evidence against	Assessment
2,167 connects equal 2,167 affected Java hosts	`fake3306c2b.log` (1,145) and `fakemysql_test.log` (1,022) log 2,167 connects.	Public TCP 3306 ports draw constant bot scans; `fake3306c2b.log` logs 1 `mysql-connector-java-8.0.28` query session, and 7 sized captures sit on disk.	**Unsupported.** 2,167 mixes internet scan hits with real JDBC callbacks.

Hypothesis	Evidence for	Evidence against	Assessment
Shared operator with the India TCP 8001 VPS	Both hosts were open on `20 November 2026` and target MySQL JDBC clients.	Comparing **267** primary hashes with **506** India VPS hashes yields **0 shared non-public hashes**, and toolsets differ (`java-chains` + `MySQL_Fake_Server`).	**Unlinked parallel host.** Nothing links the two operators.
Authorized driver security research	**89.22%** of entries (**927 / 1,039**) sit in `audit/` and `mysql_src/`.	Runs public TCP **3306** listeners and keeps **7** captured target files (`/etc/passwd`, SSH keys, Git creds, **21.59 MB** log) on an open cloud host.	**Contradicted.** Keeping third-party keys and logs on an open server breaks testing rules.
Confirmed attacks via `pgjdbc`, MSSQL, Shiro, or JNDI	Stages `src_pg/`, `pg.jar`, `sqljdbc4.jar`, `shiro-*.jar`, and `JNDIExploit-1.4-SNAPSHOT.jar`.	Stored logs record TCP **3306** MySQL traffic only; no PostgreSQL, MSSQL, Shiro, or LDAP callback logs appear.	**Staged only.** Keep staged multi-engine and JNDI files separate from observed MySQL captures.

7. Validated Detection Rules

Both server IPs, all client IPs, and all **267 + 506** SHA-256 digests are withheld. Defenders should deploy **protocol, HTTP body, and process rules** that catch this tradecraft on any host. Every rule passed checks on Suricata `7.0.12` (`suricata -T` and linter, **0** problems), Microsoft `Kusto.Language` (**0** errors), `pySigma` (**3** rules), and `YARA-X` (**4** `filesize`-bounded rules).

7.1 Suricata 7.x Network Signatures (`sid:9558001` –9558007)

Rules `sid:9558001` –**9558004** match inbound HTTP `POST` bodies holding `jdbc:mysql:` or `jdbc:postgresql:` URLs with `allowLoadLocalInfile=true`, `autoDeserialize=true`, `queryInterceptors=`, or `socketFactory=`. They need cleartext HTTP or a TLS reverse proxy, and URL-encoded form data needs WAF decoding.

Rules `sid:9558005` –**9558007** match server-to-client TCP packets (`flow:established,to_client`) holding MySQL `LOCAL INFILE` bytes `|01 fb|` for `/etc/passwd`, `.git-credentials`, or `.ssh/id_`. They inspect cleartext MySQL streams (fake servers skip TLS) and do not see TLS MySQL traffic. Because `alert` rules do not block inline, enforce blocks in the WAF or firewall.

```
SURICATA

# Odin by Brandefense - Monograph 58 (Operation FalsePort) Suricata 7.x Ruleset
# Validated against Suricata 7.0.12 (-T) and Odin Suricata linter (0 problems).

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Inbound MySQL JDBC allowLoa
dLocalInfile Tampering"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.re
quest_body; content:"jdbc:mysql:"; nocase; content:"allowLoadLocalInfile=true"; nocase; fast_patte
rn; classtype:web-application-attack; sid:9558001; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Inbound MySQL JDBC autoDese
rialize Tampering"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.request
_body; content:"jdbc:mysql:"; nocase; content:"autoDeserialize=true"; nocase; fast_pattern; classt
ype:web-application-attack; sid:9558002; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Inbound MySQL JDBC queryInt
erceptors Tampering"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.reque
st_body; content:"jdbc:mysql:"; nocase; content:"queryInterceptors="; nocase; fast_pattern; classt
ype:web-application-attack; sid:9558003; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Inbound PostgreSQL JDBC soc
ketFactory Tampering"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.requ
est_body; content:"jdbc:postgresql:"; nocase; content:"socketFactory="; nocase; fast_pattern; clas
stype:web-application-attack; sid:9558004; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Rogue MySQL Server 0xFB LOCA
L INFILE /etc/passwd Request"; flow:established,to_client; content:"|01 fb|"; content:"/etc/passw
d"; distance:0; within:64; fast_pattern; classtype:attempted-recon; sid:9558005; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Rogue MySQL Server 0xFB LOCA
L INFILE Git Credentials Request"; flow:established,to_client; content:"|01 fb|"; content:".git-cr
edentials"; distance:0; within:96; fast_pattern; classtype:policy-violation; sid:9558006; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Rogue MySQL Server 0xFB LOCA
L INFILE SSH Private Key Request"; flow:established,to_client; content:"|01 fb|"; content:".ssh/id
_"; distance:0; within:96; fast_pattern; classtype:attempted-user; sid:9558007; rev:1;)
```

7.2 Microsoft Sentinel KQL Hunt

Replace the placeholder hosts and IPs in `ApprovedAutomationHosts` and `ApprovedExternalDatabaseIPs` with your exact CI servers and external SQL hosts before scheduling.

```kql
// Odin by Brandefense - Monograph 58 (Operation FalsePort)
// Hunts Java processes initiating unapproved external TCP 3306 MySQL sessions and CLI invocations
with risky JDBC properties or rogue listener tools.
let Lookback = 7d;
let ApprovedAutomationHosts = dynamic(["app-ci-01.corp.example", "db-etl-01.corp.example"]);
let ApprovedExternalDatabaseIPs = dynamic(["198.51.100.10", "198.51.100.11"]);
let SuspiciousJdbcOrListenerProcess =
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAutomationHosts))
    | where (
        ProcessCommandLine has_any ("allowLoadLocalInfile=true", "autoDeserialize=true", "queryInt
erceptors=", "socketFactory=", "authenticationPluginClassName=")
        or ProcessCommandLine has_any ("JNDIExploit", "MySQL_Fake_Server", "java-chains", "fakemys
ql")
    )
    | project Timestamp, DeviceName, AccountName, SignalType = "RiskyJdbcOrListenerCLI", Detail =
ProcessCommandLine;
let UnapprovedJavaMySqlEgress =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAutomationHosts))
    | where InitiatingProcessFileName in~ ("java", "java.exe")
    | where RemoteIPType == "Public" and RemotePort == 3306
    | where not(RemoteIP in (ApprovedExternalDatabaseIPs))
    | project Timestamp, DeviceName, AccountName = InitiatingProcessAccountName, SignalType = "Una
pprovedJavaPublic3306Egress", Detail = strcat(InitiatingProcessFileName, " -> ", RemoteIP, ":", to
string(RemotePort));
union SuspiciousJdbcOrListenerProcess, UnapprovedJavaMySqlEgress
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), EventCount = count(), Signals =
make_set(SignalType, 6), SampleDetails = make_set(Detail, 5) by DeviceName, AccountName
| order by LastSeen desc
```

7.3 Portable Sigma and YARA Rules

- **Sigma (`sigma_falseport_jdbc_tampering_and_rogue_mysql.yml`).** Rule 1 catches `/java` CLI flags `allowLoadLocalInfile=true`, `autoDeserialize=true`, `queryInterceptors=`, `socketFactory=`, or `authenticationPluginClassName=`. Rule 2 catches `JNDIExploit`, `MySQL_Fake_Server`, `java-chains`, `fakemysql`, or `fake3306` runs. Rule 3 catches child processes of `/java` naming `/etc/passwd`, `.git-credentials`, `.ssh/id_rsa`, or `.ssh/id_ed25519`. Pair Rule 3 with `sid:9558005`–**9558007** because direct `LOCAL INFILE` reads stay inside the JVM.
- **YARA (`odin_falseport_rogue_mysql_and_jdbc_audit.yar`).** Four `filesize`-bounded generic rules match `0xFB` `LOCAL INFILE` scripts, stored rogue MySQL logs, multi-engine JDBC audit scripts, and `JNDIExploit` or `java-chains` outputs and configs.

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

No server IPv4 addresses or workbench SHA-256 digests appear in the indicator feeds because both servers hold captured target files. Defenders should hunt with the 10 protocol, JDBC, and host markers below.

Marker category	Pattern / protocol artifact	Layer	Defensive hunt & control
MySQL JDBC file-read flag	`allowLoadLocalInfile=true`	HTTP / JVM	Reject in DataSource URL checks; alert on visible HTTP `POST` bodies (`sid:9558001`).
MySQL JDBC deserialize flag	`autoDeserialize=true`	HTTP / JVM	Strip from user `jdbc:mysql:` URLs; alert via `sid:9558002` and Sigma Rule 1.
MySQL 8.x interceptor flag	`queryInterceptors=`	HTTP / JVM	Reject custom interceptors in `jdbc:mysql:` URLs; alert via `sid:9558003`.
PostgreSQL `pgjdbc` sink	`socketFactory=` (`CVE-2022-21724`)	HTTP / JVM	Upgrade `pgjdbc` to `>= 42.2.25` / `42.3.2` and alert on `POST` bodies (`sid:9558004`).
Rogue MySQL `0xFB` packet	`|01 fb|` + `/etc/passwd`	TCP 3306	Alert on cleartext server `LOCAL INFILE` requests (`sid:9558005`) and block egress.
Rogue MySQL `0xFB` packet	`|01 fb|` + `.git-credentials`	TCP 3306	Alert on cleartext `LOCAL INFILE` Git credential reads (`sid:9558006`).
Rogue MySQL `0xFB` packet	`|01 fb|` + `.ssh/id_`	TCP 3306	Alert on cleartext `LOCAL INFILE` SSH key requests (`sid:9558007`) and isolate JVM users.
Connector/J session query	`SELECT @@session.auto_increment_increment`	TCP 3306	Hunt unapproved outbound Java TCP 3306 sessions to public IPs (Sentinel KQL).
Staged JNDI archive	`JNDIExploit-1.4-SNAPSHOT.jar` (42.67 MB)	Linux EDR	Alert on CLI runs via Sentinel KQL and Sigma Rule 2.

Marker category	Pattern / protocol artifact	Layer	Defensive hunt & control
Parallel rogue MySQL suite	`MySQL_Fake_Server` / `java-chains`	Linux EDR	Sweep hosts and lab VMs for fake MySQL and Java gadget suites (Sigma Rule 2, YARA Rule 4).

9. Hardening Priorities for Java Platforms and Database Teams

Security and platform teams should close DataSource URL checks, driver patches, and egress gaps together:

1. **Lock down DataSource URLs and disable `LOCAL INFILE` (`0-48 Hours`).** Build `jdbc:` URLs from checked host, port, and database fields rather than raw user input (`[REF-09]`). Strip `allowLoadLocalInfile`, `autoDeserialize`, `queryInterceptors`, and `socketFactory`. Force `allowLoadLocalInfile=false`, `allowUrlInLocalInfile=false`, `allowLoadLocalInfileInPath=""`, and `autoDeserialize=false` in code (`[REF-03]`, `[REF-04]`).
2. **Patch MySQL Connector/J, PostgreSQL `pgjdbc`, and Shiro (`0-72 Hours`).** Upgrade MySQL Connector/J to **`8.2.0` or later**. That patch fixes `CVE-2023-22102` (`[REF-01]`) and `CVE-2021-2471` (fixed in `8.0.27`, `[REF-02]`). Upgrade `pgjdbc` to **`42.2.25` or `42.3.2` or later** (`CVE-2022-21724`, `[REF-05]`, `[REF-06]`), replace old `sqljdbc4.jar` files with `mssql-jdbc` using `encrypt=true;trustServerCertificate=false` (`[REF-07]`), and check Apache Shiro (`[REF-08]`).
3. **Block outbound SQL and JNDI ports from JVM subnets (`Days 1-7`).** Stop web and BI hosts from opening outbound TCP **3306**, **5432**, **1433**, **1099**, or **1389** sessions to public IPs. Use default-deny internet egress on JVM VLANs so custom fake MySQL ports also fail.
4. **Isolate JVM accounts from keys, Git stores, and verbose logs (`Days 7-14`).** Run Spring Boot and Java apps under dedicated `nologin` Linux users with empty home folders (`[REF-10]`). Keep `.ssh/id_*` keys and `.git-credentials` off app hosts, and scrub secrets from app logs.
5. **Deploy wire and host detections (`Days 7-30`).** Deploy Suricata `sid:9558001`–**9558007** behind TLS proxies and on egress links. Schedule the Sentinel KQL hunt and scan build hosts with the bounded YARA rules.

10. Scope, Limits, and External References

Odin by Brandefense analyzed workbench files recorded on `13-20 November 2026`. Primary counts cover all **1,039** deduplicated entries: **263** sized files (**927,357,421** bytes) plus **776** unsized entries (**485** active files and **32** folders reported on `20 November 2026`).

Inbound `== connect from` totals (**2,167** across two TCP **3306** logs) mix automated internet scan noise with real JDBC client sessions. Folder listings alone do not tie each of the seven captured files to a specific log line or count affected organizations. Both server IPv4 addresses, all **267 + 506** SHA-256 digests, client IPs, and captured filenames are withheld.

1. [REF-01] NIST NVD, *CVE-2023-22102 - Oracle MySQL Connector/J (8.1.0 and prior)* (verified on 24 November 2026).
2. [REF-02] NIST NVD, *CVE-2021-2471 - Oracle MySQL Connector/J (8.0.26 and prior)* (verified on 24 November 2026).
3. [REF-03] Oracle MySQL Documentation, *MySQL Connector/J Developer Guide: Security Connection Properties* (verified on 24 November 2026).
4. [REF-04] Oracle MySQL Documentation, *MySQL 8.0 Reference Manual: Security Considerations for LOAD DATA LOCAL* (verified on 24 November 2026).
5. [REF-05] NIST NVD, *CVE-2022-21724 - PostgreSQL JDBC Driver (pgjdbc) Unchecked Class Instantiation* (verified on 24 November 2026).
6. [REF-06] PostgreSQL JDBC (`pgjdbc`) Security Advisory, *GHSA-v7wg-cpwc-24m4: Arbitrary Class Instantiation via Connection Properties* (verified on 24 November 2026).
7. [REF-07] Microsoft Learn, *Understanding Encryption Support - Microsoft JDBC Driver for SQL Server* (verified on 24 November 2026).
8. [REF-08] Apache Software Foundation, *Apache Shiro Security Reports* (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, *Data from Local System (T1005)* (verified on 24 November 2026).

BRANDEFENSE CAPABILITIES RELATED TO THIS RESEARCH

Cyber Threat Intelligence

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

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_falseport_jdbc_tampering_and_mysql_fileread.rules` 11 lines

```
# Odin by Brandefense - Monograph 58 (Operation FalsePort) Suricata 7.x Ruleset
# Validated against Suricata 7.0.12 (-T) and Odin Suricata linter (0 problems).

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Inbound MySQL JDBC allowLoa
dLocalInfile Tampering"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.re
quest_body; content:"jdbc:mysql:"; nocase; content:"allowLoadLocalInfile=true"; nocase; fast_patte
rn; classtype:web-application-attack; sid:9558001; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Inbound MySQL JDBC autoDese
rialize Tampering"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.request
_body; content:"jdbc:mysql:"; nocase; content:"autoDeserialize=true"; nocase; fast_pattern; classt
ype:web-application-attack; sid:9558002; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Inbound MySQL JDBC queryInt
erceptors Tampering"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.reque
st_body; content:"jdbc:mysql:"; nocase; content:"queryInterceptors="; nocase; fast_pattern; classt
ype:web-application-attack; sid:9558003; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Inbound PostgreSQL JDBC soc
ketFactory Tampering"; flow:established,to_server; http.method; content:"POST"; bsize:4; http.requ
est_body; content:"jdbc:postgresql:"; nocase; content:"socketFactory="; nocase; fast_pattern; clas
stype:web-application-attack; sid:9558004; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Rogue MySQL Server 0xFB LOCA
L INFILE /etc/passwd Request"; flow:established,to_client; content:"|01 fb|"; content:"/etc/passw
d"; distance:0; within:64; fast_pattern; classtype:attempted-recon; sid:9558005; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Rogue MySQL Server 0xFB LOCA
L INFILE Git Credentials Request"; flow:established,to_client; content:"|01 fb|"; content:".git-cr
edentials"; distance:0; within:96; fast_pattern; classtype:policy-violation; sid:9558006; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R58 FalsePort Rogue MySQL Server 0xFB LOCA
L INFILE SSH Private Key Request"; flow:established,to_client; content:"|01 fb|"; content:".ssh/id
_"; distance:0; within:96; fast_pattern; classtype:attempted-user; sid:9558007; rev:1;)
```

Show full rule

YARA `odin_falseport_rogue_mysql_and_jdbc_audit.yar` 78 lines

```
/*
   Odin by Brandefense - Monograph 58 (Operation FalsePort) YARA Suite
   4 filesize-bounded structural rules covering rogue MySQL 0xFB LOCAL INFILE listeners,
   stored rogue MySQL session logs, multi-engine JDBC property audit scripts, and
   JNDIExploit / java-chains artifacts (zero victim or vendor identifiers).
*/

rule ODIN_FalsePort_Rogue_MySQL_Local_Infile_Server
{
    meta:
```

```
        author = "Odin by Brandefense"
        monograph = "58"
        description = "Detects rogue MySQL listener scripts crafting 0xFB LOCAL INFILE file-read p
ackets for sensitive Linux files"
        date = "2026-11-24"
    strings:
        $m1 = "@@session.auto_increment_increment" ascii nocase
        $m2 = "SHOW SESSION STATUS" ascii nocase
        $m3 = "mysql_native_password" ascii
        $p1 = "\\xfb" ascii nocase
        $p2 = "0xfb" ascii nocase
        $f1 = "/etc/passwd" ascii
        $f2 = ".git-credentials" ascii
        $f3 = ".ssh/id_" ascii
    condition:
        filesize < 512KB and ($m1 or $m2 or $m3) and ($p1 or $p2) and ($f1 or $f2 or $f3)
}

rule ODIN_FalsePort_Rogue_MySQL_Listener_Session_Log
{
    meta:
        author = "Odin by Brandefense"
        monograph = "58"
        description = "Detects rogue MySQL TCP 3306 listener logs recording inbound Connector/J ha
ndshakes and session queries"
        date = "2026-11-24"
    strings:
        $c1 = "== connect from" ascii
        $j1 = "mysql-connector-java" ascii
        $q1 = "@@session.auto_increment_increment" ascii
        $q2 = "SHOW SESSION STATUS" ascii
    condition:
        filesize < 5MB and $c1 and $j1 and ($q1 or $q2)
}

rule ODIN_FalsePort_MultiEngine_JDBC_Audit_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "58"
        description = "Detects multi-engine JDBC connection-property audit scripts testing MySQL a
nd PostgreSQL driver sinks"
        date = "2026-11-24"
    strings:
        $d1 = "allowLoadLocalInfile" ascii
        $d2 = "autoDeserialize" ascii
        $d3 = "queryInterceptors" ascii
        $p1 = "socketFactory" ascii
        $p2 = "authenticationPluginClassName" ascii
    condition:
        filesize < 256KB and 2 of ($d*) and ($p1 or $p2)
}

rule ODIN_FalsePort_JNDIExploit_or_JavaChains_Artifact
```

```
{
    meta:
        author = "Odin by Brandefense"
        monograph = "58"
        description = "Detects JNDIExploit or java-chains listener outputs and configuration file
s"
        date = "2026-11-24"
    strings:
        $t1 = "JNDIExploit" ascii
        $t2 = "java-chains" ascii
        $r1 = "ldap://" ascii
        $r2 = "rmi://" ascii
        $g1 = "TemplatesImpl" ascii
        $g2 = "BasicController" ascii
    condition:
        filesize < 2MB and ($t1 or $t2) and ($r1 or $r2) and ($g1 or $g2)
}
```

Sentinel KQL **sentinel_falseport_rogue_mysql_and_jdbc_hunt.kql** 26 lines

```
// Odin by Brandefense - Monograph 58 (Operation FalsePort)
// Hunts Java processes initiating unapproved external TCP 3306 MySQL sessions and CLI invocations
with risky JDBC properties or rogue listener tools.
let Lookback = 7d;
let ApprovedAutomationHosts = dynamic(["app-ci-01.corp.example", "db-etl-01.corp.example"]);
let ApprovedExternalDatabaseIPs = dynamic(["198.51.100.10", "198.51.100.11"]);
let SuspiciousJdbcOrListenerProcess =
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAutomationHosts))
    | where (
        ProcessCommandLine has_any ("allowLoadLocalInfile=true", "autoDeserialize=true", "queryInt
erceptors=", "socketFactory=", "authenticationPluginClassName=")
        or ProcessCommandLine has_any ("JNDIExploit", "MySQL_Fake_Server", "java-chains", "fakemys
ql")
    )
    | project Timestamp, DeviceName, AccountName, SignalType = "RiskyJdbcOrListenerCLI", Detail =
ProcessCommandLine;
let UnapprovedJavaMySqlEgress =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAutomationHosts))
    | where InitiatingProcessFileName in~ ("java", "java.exe")
    | where RemoteIPType == "Public" and RemotePort == 3306
    | where not(RemoteIP in (ApprovedExternalDatabaseIPs))
    | project Timestamp, DeviceName, AccountName = InitiatingProcessAccountName, SignalType = "Una
pprovedJavaPublic3306Egress", Detail = strcat(InitiatingProcessFileName, " -> ", RemoteIP, ":", to
string(RemotePort));
union SuspiciousJdbcOrListenerProcess, UnapprovedJavaMySqlEgress
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), EventCount = count(), Signals =
```

```
make_set(SignalType, 6), SampleDetails = make_set(Detail, 5) by DeviceName, AccountName
| order by LastSeen desc
```

Sigma `sigma_falseport_jdbc_tampering_and_rogue_mysql.yml` 87 lines

```
title: FalsePort Java Process with Risky JDBC File-Read or Interceptor Properties
id: 8c58a1b2-3d41-4e79-9b14-58c10f92a301
status: experimental
description: Detects Java command lines specifying dangerous MySQL Connector/J or PostgreSQL pgjdb
c connection properties associated with client file reads or arbitrary class instantiation.
author: Odin by Brandefense
date: 2026-11-24
logsource:
  category: process_creation
  product: linux
detection:
  selection_java:
    Image|endswith: '/java'
  selection_props:
    CommandLine|contains:
      - 'allowLoadLocalInfile=true'
      - 'autoDeserialize=true'
      - 'queryInterceptors='
      - 'socketFactory='
      - 'authenticationPluginClassName='
  filter_ci_hosts:
    ComputerName:
      - 'app-ci-01.corp.example'
  condition: selection_java and selection_props and not filter_ci_hosts
falsepositives:
  - Controlled driver security testing on designated integration build hosts
level: high
tags:
  - attack.initial-access
  - attack.t1190
---
title: FalsePort Rogue MySQL Listener or JNDIExploit Framework Invocation
id: 8c58a1b2-3d41-4e79-9b14-58c10f92a302
status: experimental
description: Detects process execution of JNDIExploit, java-chains, MySQL_Fake_Server, or custom r
ogue MySQL listener scripts on Linux servers.
author: Odin by Brandefense
date: 2026-11-24
logsource:
  category: process_creation
  product: linux
detection:
  selection_tools:
    CommandLine|contains:
      - 'JNDIExploit'
      - 'MySQL_Fake_Server'
      - 'java-chains'
```

```
      - 'fakemysql'
      - 'fake3306'
  filter_sec_lab:
    ComputerName:
      - 'sec-lab-01.corp.example'
  condition: selection_tools and not filter_sec_lab
falsepositives:
  - Authorized security research environments running on isolated lab hosts
level: critical
tags:
  - attack.initial-access
  - attack.t1190
---
title: FalsePort Shell or Python Access to Git Credentials and SSH Keys Spawned by Java
id: 8c58a1b2-3d41-4e79-9b14-58c10f92a303
status: experimental
description: Detects child processes spawned by a Java application server referencing /etc/passwd,
.git-credentials, or .ssh/id_ key files on the command line.
author: Odin by Brandefense
date: 2026-11-24
logsource:
  category: process_creation
  product: linux
detection:
  selection_parent:
    ParentImage|endswith: '/java'
  selection_targets:
    CommandLine|contains:
      - '/etc/passwd'
      - '.git-credentials'
      - '.ssh/id_rsa'
      - '.ssh/id_ed25519'
  filter_ci_hosts:
    ComputerName:
      - 'app-ci-01.corp.example'
  condition: selection_parent and selection_targets and not filter_ci_hosts
falsepositives:
  - Build-agent Java runners such as Jenkins agents managing approved repository credentials
level: high
tags:
  - attack.collection
  - attack.t1005
```

