New issue

Path Traversal via Untrusted MQTT Messages Leads to Directory Enumeration in Android Client #25

Open

AnalogyC0de opened 3 weeks ago · edited by AnalogyC0de Edits Owner

Path Traversal via Untrusted MQTT Messages Leads to Directory Enumeration in Android Client

Project Information

- **Project:** FedML
- **Component:** Android Training Client (fedmlsdk)
- **Repository:** https://github.com/FedML-AI/FedML
- **Affected Versions:** <=0.8.9

Vulnerability Summary

A **path traversal vulnerability (CWE-22)** exists in the Android client of FedML.
The client processes **MQTT messages as task instructions** and uses the `dataSet` parameter to construct filesystem paths **without validation**.

An attacker who can **publish or tamper with MQTT messages** can supply crafted path traversal payloads (e.g., `../../../../`) to cause the client to access and enumerate arbitrary directories within the app's accessible filesystem.

Affected Components

File	Description
`FileUtils.java:7`	Directory listing sink (`File.list()`)
`TrainingExecutor.java:63`	Path construction using dataset
`ClientManager.java:77`	Message handling
`ClientAgentManager.java:98`	MQTT message processing

Technical Details

Data Flow

```
MQTT Message (dataSet)
        ↓
ClientAgentManager / ClientManager
        ↓
TrainingExecutor (trainDataPath construction)
        ↓
FileUtils (File.list())
        ↓
Directory Enumeration (SINK)
```

Root Cause

- **Untrusted input:** `dataSet` from MQTT message
- **No validation or sanitization**
- **Direct use in filesystem path construction**

Vulnerable Pattern

```
String trainDataPath = basePath + "/" + dataSet;
File dir = new File(trainDataPath);
String[] files = dir.list();   // SINK
```

Issue

- No canonicalization (`getCanonicalPath`)
- No traversal filtering (`../`)
- No base path enforcement

Proof of Concept (PoC)

Malicious MQTT Payload

```
{
  "dataSet": "../../../../data/data/"
}
```

Attack Steps

1. Attacker gains ability to publish MQTT messages (see threat model below)
2. Sends crafted message with traversal payload
3. Android client processes message
4. Client lists unintended directory

Example Result (Observed Behavior)

```
/data/data/
├── ai.fedml.app/
│   ├── files/
│   ├── cache/
│   ├── databases/
```

Threat Model (Critical for Validity)

This vulnerability assumes **MQTT is not fully trusted**, which is realistic in many deployments.

Attack Preconditions (any of the following)

- MQTT broker lacks strong authentication
- Topic ACLs are not strictly enforced
- Network attacker can perform MITM
- Compromised or rogue server sends malicious tasks

Security Boundary Crossed

```
Network (MQTT) → Local Filesystem (Android sandbox)
```

This represents a **clear trust boundary violation**

Impact

1. Directory Enumeration (Primary Impact)

- Lists contents of app-accessible directories
- Enables filesystem reconnaissance

---

2. Application Fingerprinting

- Identify installed components
- Infer app structure and storage layout

---

3. Sensitive Path Disclosure

- Training dataset locations
- Internal storage structure

---

4. Potential Denial of Service

- Large directory traversal may cause:

- High memory usage
- UI freeze / ANR

Limitations

- No direct file read (only listing)
- Limited to app-accessible filesystem (Android sandbox)

Why This Is a Valid Vulnerability

- Input is **remotely controllable (MQTT)**
- Crosses **trust boundary (network → filesystem)**
- No validation or restriction
- Leads to **information disclosure**

This satisfies **CWE-22 (Path Traversal)** conditions

Remediation

1. Input Validation (Required)

```
if (dataSet.contains("..")) {
    throw new SecurityException("Invalid dataset path");
}
```

---

2. Canonical Path Enforcement

```
File baseDir = new File(BASE_PATH);
File target = new File(baseDir, dataSet);

if (!target.getCanonicalPath().startsWith(baseDir.getCanonicalPath())) {
    throw new SecurityException("Path traversal detected");
}
```

---

3. Restrict Allowed Dataset Names

```
// Example: whitelist pattern
if (!dataSet.matches("^[a-zA-Z0-9_\\-]+$")) {
    throw new SecurityException("Invalid dataset name");
}
```

---

4. Harden MQTT Security

- Enforce authentication (TLS + client certs)
- Apply strict topic ACLs
- Prevent unauthorized publishers

Verification Steps

Test 1: Traversal Payload

```
dataSet = "../../../../data/data/"
```

Expected (after fix):

```
Error: Path traversal detected
```

Test 2: Valid Dataset

```
dataSet = "mnist"
```

Expected:

```
Directory listing succeeds
```

Final Assessment

> This vulnerability allows an attacker with access to the MQTT communication channel to perform path traversal and enumerate directories on the Android client. Due to the lack of input validation and the crossing of a network-to-filesystem trust boundary, this issue constitutes a valid CWE-22 vulnerability with moderate impact.

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

No branches or pull requests

Participants