Rimini Street

Engineered for Support™

Security Vulnerability Analysis Report

SAP NetWeaver Visual Composer Unrestricted File Upload Vulnerability (0-day)

CVE-2025-31324

April 25, 2025

Contents

SAP NetWeaver Visual Composer Unrestricted File Upload Vulnerability (0-day)			
Vulnerability ID	CVE-2025-31324	**CVSS Score**	10.0
Affected Product	SAP NetWeaver Visual Composer	**Impact Score**	6.0
Affected Component(s)	EP-VC-INF	**Exploitability Score**	3.9
Affected Version(s)	VCFRAMEWORK 7.50	**EPSS Score(%)**	0.04
Severity	Critical	**Temporal Score**	8.8
Remotely Exploitable w/o Authentication?	Yes	**Attack Vector**	Network
User Interaction	None	**Scope**	Changed

Rimini Protect™ Security Services has observed that an Unrestricted File Upload vulnerability in SAP NetWeaver Java (CVE-2025-31324) is actively being actively exploited in the wild.

This vulnerability exists in the Metadata Uploader component of SAP NetWeaver Visual Composer and has received a CVSS score of 10.0 (Critical), indicating the maximum possible risk level.

The authorization mechanism of the Metadata Uploader fails to properly validate that users have the appropriate permissions before allowing file uploads. As a result, unauthenticated attackers can upload executable files into public folders, potentially causing a full system compromise, including unrestricted access to the SAP business data and processes, deploy ransomware in SAP and move laterally.

Note: SAP released an emergency security note (# **3594142**) on April 24th to address this issue.

Vulnerability Details

SAP NetWeaver Visual Composer Metadata Uploader is not protected with a proper authorization. This oversight allows an unauthenticated agent to bypass security controls and upload potentially malicious executable binaries that could severely harm the host system. This could significantly affect the confidentiality, integrity, and availability of the targeted system.

Prerequisites

This issue is affects Visual Composer which is a development tool in SAP NetWeaver.

Visual Composer is considered deprecated and is **not** enabled by default.

Therefore, unless this tool is enabled, your system is not affected by this vulnerability.

Diagnostics

Vulnerable URL endpoint/path is **/developmentserver/metadatauploader.**

To check if your system is vulnerable, follow the steps below:

1. Test if the following URL is accessible without authentication: *https://[your-sap-server]/developmentserver/metadatauploader*
2. If you can access this page without being prompted for credentials, your system is vulnerable.

Impact

Organizations that rely heavily on SAP solutions, including manufacturing, healthcare, financial services, and critical infrastructure, are at risk from this vulnerability.

According to this LinkedIn post:

- Attackers are leveraging the **metadatauploader** endpoint to deploy JSP webshells in accessible files and directories via crafted POST requests, and then execute them via simple GET requests, obtaining full control of the vulnerable endpoint.
- Tools like Brute Ratel C4 and Heaven's Gate are being used to maintain stealthy control and persistence.
- Some intrusions show delayed post-exploitation, indicating likely use by Initial Access Brokers (IABs).

In all cases, the JSP webshells were planted in the same root directory, had similar functionality, and shared code from a public GitHub repository on remote code execution (RCE) via file uploads.

According to ReliaQuest, the unauthorized file upload and execution activities appeared linked to the exploitation of CVE-2017-9844 (CVSS score - 9.8), a Metadata Uploader vulnerability that could result in Denial of Service (DoS) conditions and code execution via crafted serialized Java objects.

Mitigation/Hardening

Restrict access to vulnerable URL endpoint: **/developmentserver/metadatauploader.** Use firewall, access control list or other network tools, as appropriate, to restrict access.

To ensure the system is not exploitable, disable the vulnerable application. Follow the steps below:

1. Log on to the NetWeaver Administration (NWA) tool.
2. Access **HTTP Provider Configuration**.
3. Open the **Application Aliases** tab for each virtual host.
4. Look for the alias **developmentserver**.
5. **De-select** the checkbox under the **Active** column.
6. Save settings.

If Visual Composer is required for development work, ensure that it is active only in development environment. You can disable Visual Composer as follows:

1. Log on to the NetWeaver Administration (NWA) tool.
2. Click **Operation**.
3. Click **Start and Stop** and then click **JAVA Application**.
4. Click **More Actions** and then choose **Edit Startup Filters**.
5. Click **Add** and add the following details:
   Action: Disable
   Component: Service
   Component Name Mask: developmentserver
6. Click **Set** and then click **Save**.
7. Finally, **Restart the Cluster**.

Additional Recommendations

Here are a few other recommendations:

- Look for unauthorized access attempts to the **/developmentserver/metadatauploader** path. Check for patterns like:
  - POST requests to this endpoint, especially from external IP addresses.
  - Successful requests (HTTP 200 status codes) that were not authenticated.
  - Any file upload activity to this path.
- Check all SAP NetWeaver instances for:
  - Unexpected JSP uploads in servlet directories.
  - Brute Ratel signatures.
  - Memory tampering via NtSetContextThread.
- Check web server logs for any suspicious activity, such as attempts to upload or access unauthorized files.
- Monitor for unauthorized outbound connections from your SAP systems.
- Make sure the SAP NetWeaver systems are within securely segmented networks.
- Implement strict access controls to limit access to sensitive files and directories on SAP NetWeaver. Make sure only authenticated users have upload permissions to SAP components.

Rimini Protect™ Application Risk Mitigation

If the environment is exposed externally, then Rimini Protect™ Application Risk Mitigation (ARM), powered by RedShield, can provide protection against this and similar CVEs from being exploited via delivered Rules/Shields. This is a completely managed service. Contact your Senior Engagement Manager (SEM) to get additional details.

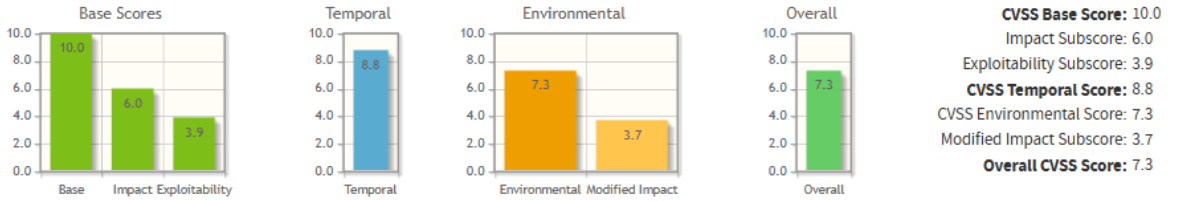
If you are utilizing Rimini Protect™ Application Risk Mitigation, you are protected. No action is required.

Common Vulnerability Scoring System Calculator for CVE-2025-31324

Rimini Street recommends utilizing the NIST CVSS Calculator to evaluate risk to this vulnerability.

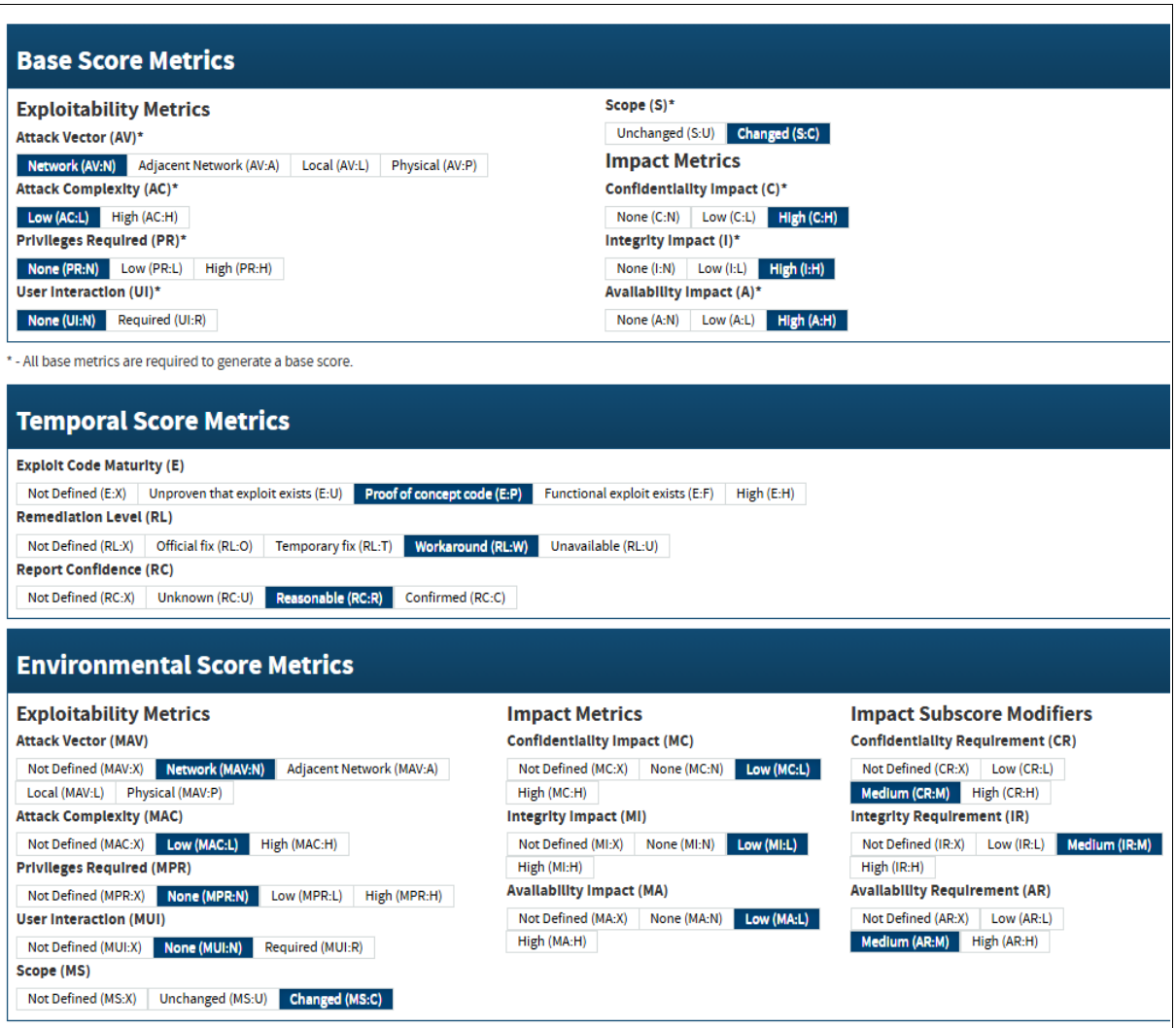
The Base Score of this vulnerability is **10.0**. By updating the Temporal and Environmental scores in the calculator to reflect access controls and restrictions to the environment, the Overall Risk Score is reduced to **7.3**.

Please refer to the following screenshots. Rimini Street is happy to help determine your residual risk level and further apply corrective controls should your Overall Risk Score be outside of your accepted risk tolerance.



CVSS v3.1 Vector

AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H/E:P/RL:W/RC:R/CR:M/IR:M/AR:M/MAV:N/MAC:L/MPR:N/MUI:N/MS:C/MC:L/MI:L/MA:L



Additional Information

The relevant weakness type for CVE-2025-31324 is CWE-434: Unrestricted Upload of File with Dangerous Type (**Unrestricted File Upload**).

This weakness occurs when the product allows the upload or transfer of dangerous file types that are automatically processed within its environment. Here are a few hardening guidelines to mitigate CWE-434:

- Generate a new, unique filename for an uploaded file instead of using the user-supplied filename, so that no external input is used at all.
- When the set of acceptable objects, such as filenames or URLs, is limited or known, create a mapping from a set of fixed input values (such as numeric IDs) to the actual filenames or URLs, and reject all other inputs.
- Consider storing the uploaded files outside of the web document root entirely. Then, use other mechanisms to deliver the files dynamically.
- Define a very limited set of allowable extensions and only generate filenames that end in these extensions. Consider the possibility of Cross-Site Scripting (CWE-79) before allowing .html or .htm file types.

- Ensure that only one extension is used in the filename. Some web servers, including some versions of Apache, may process files based on inner extensions so that filename.php.gif is fed to the PHP interpreter.
- When running on a web server that supports case-insensitive filenames, perform case-insensitive evaluations of the extensions that are provided.
- For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.
- Do not rely exclusively on sanity checks of file contents to ensure that the file is of the expected type and size. It may be possible for an attacker to hide code in some file segments that will still be executed by the server. For example, GIF images may contain a free-form comments field.
- Do not rely exclusively on the MIME content type or filename attribute when determining how to render a file. Validating the MIME content type and ensuring that it matches the extension is only a partial solution.

References

- National Vulnerability Database
- Onapsis
- RedRays
- ReliaQuest
- The Hacker News
- Security Week
- LinkedIn Post