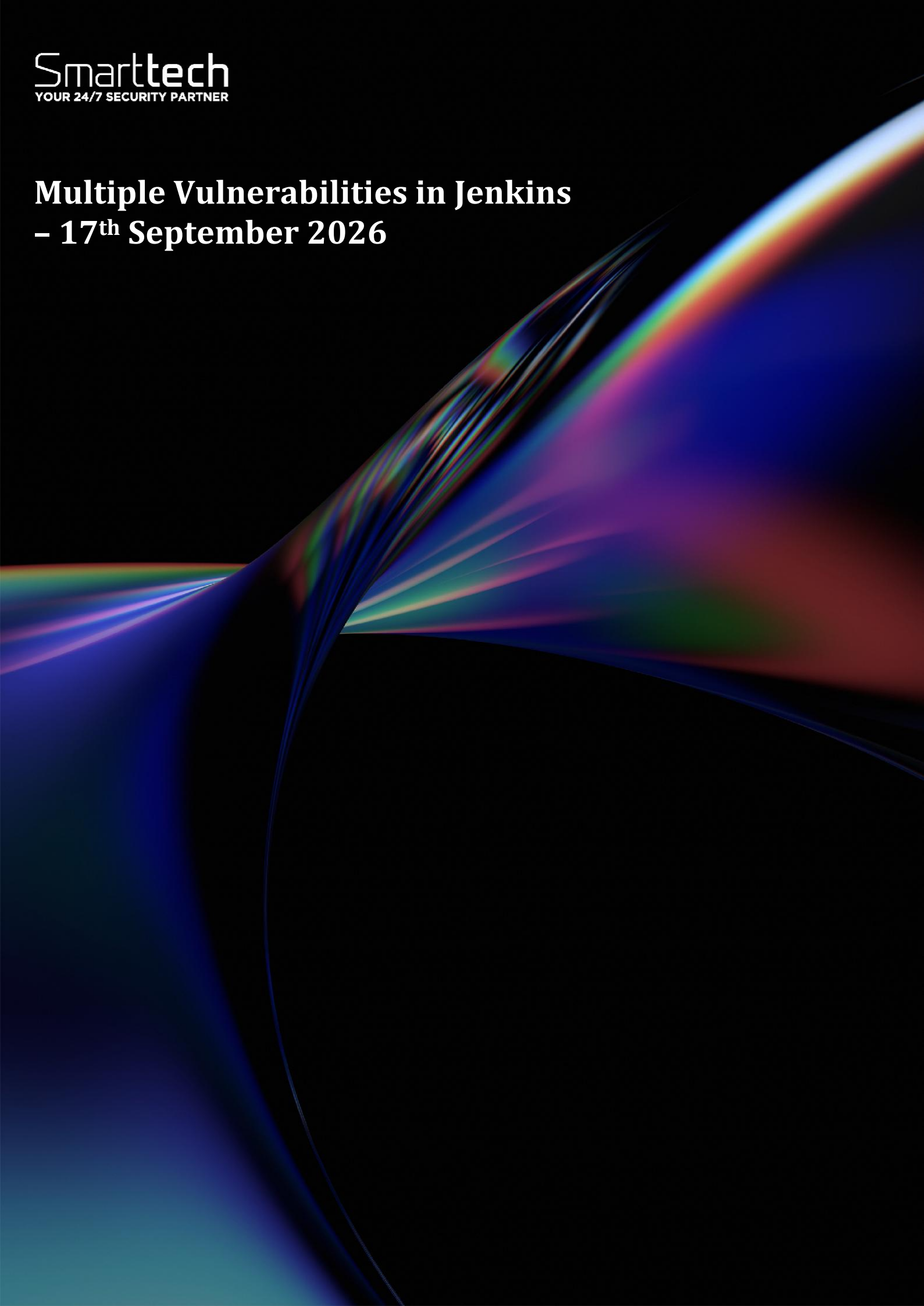
Smarttech
YOUR 24/7 SECURITY PARTNER

# Multiple Vulnerabilities in Jenkins – 17th September 2026



| | |
|---|---|
| **Document ID** | SMA-Threat Report |
| **Document status** | ISSUED |
| **Issue Number** | 167 |
| **Authors** | Dorin Constantin Banu < constantin.banu@smarttech247.com > |
| **Verified by** | Alin Curcan < alin.curcan@smarttech247.com > |
| **Last modified** | 2026-09-17 |
| **Issue Date** | 2026-09-16 |

**Threat Reports** are reports created by Smarttech247 based on high and critical severity vulnerabilities that may have a high potential to be exploited in the wild i.e. vulnerabilities that are present in most used products by companies and do not have an auto-update option or they are usually not automatically updated in case that could lead to some service disruption. This report is usually created as soon as the vulnerability is released, therefore we strongly recommend that the information is reviewed, tests are performed and patches are applied before the first proof-of-concept is released. Even though certain vulnerabilities may not have an active exploit in the wild at the time that we report on them, we take into consideration the wider risk and the impact it could have on systems, should an exploit like that be available after a while. Our duty is to report them on time and we recommend enterprises that, in order to keep critical business systems protected, they should consider, on average, ten working days to check whether or not the new vulnerability affects them, and if so, to implement actions in order to remove the risk.

## Overview:

Multiple vulnerabilities have been identified in Jenkins. Successful exploitation of these vulnerabilities could lead to arbitrary code execution, information disclosure, server-side request forgery (SSRF), cross-site scripting (XSS), and file-system manipulation.

### Risk

Government:

- Large and medium government entities: **High**
- Small government entities: **High**

Businesses:

- Large and medium business entities: **High**
- Small business entities: **High**

## Technical summary

More details related to these vulnerabilities are as follows:

| ***CVE ID*** | ***Description*** |
|---|---|
| Sandbox bypass vulnerability in Script Security Plugin<br>**CVE-2026-92122**<br>CVSS Base Score: 8.8 | Script Security Plugin provides a sandbox feature that allows running user-provided scripts safely by intercepting and checking potentially unsafe operations.<br>Groovy coerces a value to an interface by creating a proxy that forwards each interface method call to a method of the same name on that value.<br>Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier does not check the method that the proxy calls when the value inherits a method of the same name as an interface method.<br>This allows attackers with permission to define and run sandboxed scripts, including Pipelines, to bypass the sandbox protection and execute arbitrary code in the context of the Jenkins controller JVM. |
| Sandbox bypass vulnerability in Script Security Plugin<br>**CVE-2026-92123**<br>CVSS Base Score: 8.8 | Script Security Plugin provides a sandbox feature that allows running user-provided scripts safely by intercepting and checking potentially unsafe operations.<br>Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier does not intercept operations performed on |

| | |
|---|---|
| | a null receiver (method calls, property and attribute accesses, and array accesses).<br>This allows attackers with permission to define and run sandboxed scripts, including Pipelines, to bypass the sandbox protection and execute arbitrary code in the context of the Jenkins controller JVM. |
| Sandbox bypass vulnerability in Script Security Plugin<br>**CVE-2026-92124**<br>CVSS Base Score: **8.8** | Script Security Plugin provides a sandbox feature that allows running user-provided scripts safely by intercepting and checking potentially unsafe operations.<br>Groovy casts a collection to another type using the elements of that collection, either passing them to a constructor of that type or casting each of them.<br>Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier checks the operations Groovy will perform with the elements it reads from the collection, but performs the cast on the collection itself rather than on the elements it checked.<br>This allows attackers with permission to define and run sandboxed scripts, including Pipelines, to have the cast performed on different elements by overriding the methods of a collection, bypassing the sandbox protection and executing arbitrary code in the context of the Jenkins controller JVM. |
| Sandbox bypass vulnerability in Script Security Plugin<br>**CVE-2026-92125**<br>CVSS Base Score: 8.8 | Script Security Plugin provides a sandbox feature that allows running user-provided scripts safely by intercepting and checking potentially unsafe operations.<br>Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier does not reject the @GroovyASTTransformationClass annotation, which a script can use to link an annotation type it declares to an arbitrary AST transformation, causing Groovy to run that transformation at compile time, before the sandbox is applied.<br>This allows attackers with permission to define and run sandboxed scripts, including Pipelines, to bypass the sandbox protection and execute arbitrary code in the context of the Jenkins controller JVM. |
| Sandbox bypass vulnerability in Script Security Plugin<br>**CVE-2026-92126**<br>CVSS Base Score: **7.5** | Script Security Plugin provides a sandbox feature that allows running user-provided scripts safely by intercepting and checking potentially unsafe operations.<br>Groovy's @Builder annotation generates builder code at compile time using the strategy class named by its builderStrategy member, which can be one of the strategies provided by Groovy or a custom implementation.<br>Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier does not reject @Builder annotations whose builderStrategy member names an arbitrary class, which causes Groovy to instantiate that class at compile time, before the sandbox is applied.<br>This may allow attackers with permission to define and run sandboxed scripts, including Pipelines, to execute code outside the sandbox, in the rare case |

| | |
|---|---|
| | that a suitable class is present on the classpath of the component that evaluates the script. |
| Classpath approval bypass vulnerability in Script Security Plugin<br>**CVE-2026-92127**<br>CVSS Base Score: **8.0** | Script Security Plugin allows Groovy scripts to load classpath entries from JAR files, which administrators must approve before any script is allowed to use them.<br>Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier automatically approves the classpath entries in an item configuration when a user with Overall/Administer permission copies the item, or updates that configuration through the REST API or CLI, allowing attackers able to define classpath entries to execute arbitrary code in the context of the Jenkins controller JVM. |
| TOCTOU race condition in Script Security Plugin<br>**CVE-2026-92128**<br>CVSS Base Score: **7.5** | Script Security Plugin allows Groovy scripts to load classpath entries from JAR files, specified either by file path or by URL, which administrators must approve before any script is allowed to use them.<br>Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier downloads a JAR file specified by URL twice when a script is evaluated, confirming the approval of the first download and loading the classpath entries from the second.<br>This results in a time-of-check to time-of-use (TOCTOU) race condition that allows attackers able to define classpath entries to execute arbitrary code in the context of the Jenkins controller JVM. |
| Sandbox bypass vulnerability in Script Security Plugin<br>**CVE-2026-92129**<br>CVSS Base Score: **7.5** | Script Security Plugin provides a sandbox feature that allows running user-provided scripts safely by intercepting and checking potentially unsafe operations.<br>Groovy allows an existing class to be extended at runtime with additional methods.<br>Script Security Plugin 1415.v9a_f9b_3a_c253d and earlier does not check calls to such methods from sandboxed scripts.<br>This allows attackers with permission to define and run sandboxed scripts, including Pipelines, to bypass the sandbox protection and execute code outside the sandbox. |
| Exposure of System-scoped credentials in Pipeline: Multibranch Plugin<br>**CVE-2026-92130**<br>CVSS Base Score: **3.1** | Pipeline: Multibranch Plugin 841.vec5b_9e1806ec and earlier does not set the appropriate context for credentials lookup in the resolveScm Pipeline step, allowing the use of System-scoped credentials otherwise reserved for the global configuration.<br>This allows attackers with Item/Configure permission to access and capture credentials they are not entitled to use. |
| Path traversal vulnerability in Pipeline: Groovy Libraries Plugin<br>**CVE-2026-92131**<br>CVSS Base Score: **4.2** | Pipeline: Groovy Libraries Plugin provides the library step to retrieve a shared library from an SCM, using a library path to specify the directory containing the library inside the SCM checkout.<br>Pipeline: Groovy Libraries Plugin 805.va_fc79344957d and earlier does not restrict that library path to a relative path inside the SCM checkout. |

| | |
|---|---|
| | Additionally, it follows symbolic links to locations outside of the SCM checkout when retrieving the library.<br>This results in a path traversal vulnerability, allowing attackers able to configure Pipelines to read files in a resources directory and to delete files in a test directory on the Jenkins controller file system. |
| SSRF vulnerability in Gradle Plugin allows capturing the Develocity access key<br>**CVE-2026-92132**<br>CVSS Base Score: **5.4** | Gradle Plugin provides global options for administrators to detect build scan links in build logs, and to show an enriched build scan summary for those build scans on the build page.<br>In Gradle Plugin 2.19.1252.v15196b_5a_6e10 and earlier, the enriched build scan summary requests data from the build scan link detected in the build log, even when a Develocity server URL is configured in the global configuration.<br>This allows attackers able to control the build log to capture the Develocity access key configured in the global configuration by having Jenkins connect to an attacker-specified URL. |
| Cache confusion in GitLab Plugin<br>**CVE-2026-92133**<br>CVSS Base Score: **4.3** | GitLab Plugin allows jobs to override the globally configured GitLab API token credentials with an alternative one, and caches the GitLab API client it builds for those credentials.<br>GitLab Plugin 1.2149.vcfc32c82b_f7f and earlier derives the cache key from the credentials ID alone, omitting the folder in which the credentials are resolved, resulting in the GitLab API token credentials being taken from the folder of the job that created the cache entry.<br>This allows attackers with Item/Configure permission to access GitLab API token credentials they are not entitled to use. |
| Stored XSS vulnerability in Warnings Plugin<br>**CVE-2026-92134**<br>CVSS Base Score: **8.0** | Warnings Plugin uses analysis results IDs to create the links to analysis results on the Jenkins UI.<br>Warnings Plugin 13.10258.va_17d49a_78c3b_ and earlier does not validate the analysis results ID when a job configuration is submitted through the REST API. This allows attackers with Item/Configure permission to use a javascript: scheme URL as identifier, resulting in a stored cross-site scripting (XSS) vulnerability. |
| Stored XSS vulnerability in Coverage Plugin<br>**CVE-2026-92135**<br>CVSS Base Score: **8.0** | Coverage Plugin uses coverage results IDs to create the links to coverage results on the Jenkins UI.<br>Coverage Plugin 3.3358.v9487dde48783 and earlier does not validate the coverage results ID when a job configuration is submitted through the REST API. This allows attackers with Item/Configure permission to use a javascript: scheme URL as identifier, resulting in a stored cross-site scripting (XSS) vulnerability. |
| Stored XSS vulnerability in OWASP Dependency-Check Plugin<br>**CVE-2026-92136**<br>CVSS Base Score: **8.0** | OWASP Dependency-Check Plugin 5.6.4 and earlier does not escape CWE values from Dependency-Check reports on the Jenkins UI. |

| | |
|---|---|
| | This results in a stored cross-site scripting (XSS) vulnerability exploitable by attackers with Item/Configure permission. |
| Path traversal vulnerability in Robot Framework Plugin<br>**CVE-2026-92137**<br>CVSS Base Score: **8.8** | Robot Framework Plugin 6.2.2 and earlier does not check that the archive directory configured for Robot Framework report files is contained within the build directory on the Jenkins controller.<br>This allows attackers with Item/Configure permission to create or replace arbitrary files on the Jenkins controller file system with attacker-specified content, which can lead to remote code execution. |
| OAuth access token hijacking via callback URL manipulation in Bitbucket Server Integration Plugin<br>**CVE-2026-92138**<br>CVSS Base Score: **4.2** | The OAuth authorization endpoint in Bitbucket Server Integration Plugin 6.0.1 and earlier reads the oauth_callback URL from the submitted form rather than from the server-side stored request token. An attacker who can manipulate the form submission can cause Jenkins to redirect the authorizing user's browser to an arbitrary URL. The redirect includes the oauth_token and oauth_verifier parameters, allowing the attacker to complete the OAuth flow and obtain an access token on behalf of the victim. |
| SSRF vulnerability in Bitbucket Push and Pull Request Plugin allows capturing credentials<br>**CVE-2026-92139**<br>CVSS Base Score: **6.5** | Bitbucket Push and Pull Request Plugin provides a webhook endpoint at /bitbucket-hook/ to receive webhook notifications.<br>When acting on these notifications, Bitbucket Push and Pull Request Plugin 4.0.1 and earlier trusts values provided in the webhook payload, including certain URLs, and uses configured Bitbucket credentials to connect to those URLs.<br>This allows attackers to capture Bitbucket credentials stored in Jenkins by sending a crafted webhook payload. |
| Stored XSS vulnerability in Gitee Plugin<br>**CVE-2026-92140**<br>CVSS Base Score: **8.0** | Gitee Plugin 1301.v8957053c7902 and earlier does not escape the sender name from Gitee push webhook payloads in build causes.<br>This results in a stored cross-site scripting (XSS) vulnerability exploitable by attackers able to trigger builds via the Gitee Plugin webhook endpoint. |
| Open redirect vulnerability in Keycloak Authentication Plugin<br>**CVE-2026-92141**<br>CVSS Base Score: **4.3** | Keycloak Authentication Plugin 2.4.1 and earlier does not restrict the redirect URL after login.<br>This allows attackers to perform phishing attacks by having users go to a Jenkins URL that will forward them to a different site after successful authentication. |

## Affected Products:

- Bitbucket Push and Pull Request Plugin up to and including 4.0.1
- Bitbucket Server Integration Plugin up to and including 6.0.1
- Coverage Plugin up to and including 3.3358.v9487dde48783
- Gitee Plugin up to and including 1301.v8957053c7902
- GitLab Plugin up to and including 1.2149.vcfc32c82b_f7f
- Gradle Plugin up to and including 2.19.1252.v15196b_5a_6e10
- Keycloak Authentication Plugin up to and including 2.4.1
- OWASP Dependency-Check Plugin up to and including 5.6.4
- Pipeline: Groovy Libraries Plugin up to and including 805.va_fc79344957d

- Pipeline: Multibranch Plugin up to and including 841.vec5b_9e1806ec
- Robot Framework Plugin up to and including 6.2.2
- Script Security Plugin up to and including 1415.v9a_f9b_3a_c253d
- Warnings Plugin up to and including 13.10258.va_17d49a_78c3b_

## Fixed Versions:

- Bitbucket Push and Pull Request Plugin should be updated to version 4.1.0
- Bitbucket Server Integration Plugin should be updated to version 6.0.2
- Coverage Plugin should be updated to version 3.3361.v0626103a_67e6
- Gitee Plugin should be updated to version 1304.v2702f1d71cde
- GitLab Plugin should be updated to version 1.2152.veec0897048b_0
- Gradle Plugin should be updated to version 2.20.1253.vc116f0763a_eb_
- Keycloak Authentication Plugin should be updated to version 2.4.2
- OWASP Dependency-Check Plugin should be updated to version 5.6.5
- Pipeline: Groovy Libraries Plugin should be updated to version 806.v408277b_33d1d
- Pipeline: Multibranch Plugin should be updated to version 842.v3a_b_59b_57b_e6e
- Robot Framework Plugin should be updated to version 6.3.0
- Script Security Plugin should be updated to version 1422.v06869826dd9b_
- Warnings Plugin should be updated to version 13.10259.v80f407cb_03a_e

## Recommendations

**Smarttech247** team recommend the following actions to be taken:

- Apply appropriate updates provided by Jenkins to vulnerable systems immediately after appropriate testing. (M1051: Update Software)
    - Safeguard 7.1 : Establish and Maintain a Vulnerability Management Process: Establish and maintain a documented vulnerability management process for enterprise assets. Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.
    - Safeguard 7.2: Establish and Maintain a Remediation Process: Establish and maintain a risk-based remediation strategy documented in a remediation process, with monthly, or more frequent, reviews.
    - Safeguard 7.4: Perform Automated Application Patch Management: Perform application updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.
    - Safeguard 7.5 : Perform Automated Vulnerability Scans of Internal Enterprise Assets: Perform automated vulnerability scans of internal enterprise assets on a quarterly, or more frequent, basis. Conduct both authenticated and unauthenticated scans, using a SCAP-compliant vulnerability scanning tool.
    - Safeguard 7.7: Remediate Detected Vulnerabilities: Remediate detected vulnerabilities in software through processes and tooling on a monthly, or more frequent, basis, based on the remediation process.
    - Safeguard 12.1: Ensure Network Infrastructure is Up-to-Date: Ensure network infrastructure is kept up-to-date. Example implementations include running the latest stable release of software and/or using currently supported network-as-a-service (NaaS) offerings. Review software versions monthly, or more frequently, to verify software support.
    - Safeguard 18.1: Establish and Maintain a Penetration Testing Program: Establish and maintain a penetration testing program appropriate to the size, complexity, and

maturity of the enterprise. Penetration testing program characteristics include scope, such as network, web application, Application Programming Interface (API), hosted services, and physical premise controls; frequency; limitations, such as acceptable hours, and excluded attack types; point of contact information; remediation, such as how findings will be routed internally; and retrospective requirements.
    - Safeguard 18.2: Perform Periodic External Penetration Tests: Perform periodic external penetration tests based on program requirements, no less than annually. External penetration testing must include enterprise and environmental reconnaissance to detect exploitable information. Penetration testing requires specialized skills and experience and must be conducted through a qualified party. The testing may be clear box or opaque box.
    - Safeguard 18.3: Remediate Penetration Test Findings: Remediate penetration test findings based on the enterprise's policy for remediation scope and prioritization.
- Vulnerability scanning is used to find potentially exploitable software vulnerabilities to remediate them. (M1016: Vulnerability Scanning)
    - Safeguard 16.13: Conduct Application Penetration Testing: Conduct application penetration testing. For critical applications, authenticated penetration testing is better suited to finding business logic vulnerabilities than code scanning and automated security testing. Penetration testing relies on the skill of the tester to manually manipulate an application as an authenticated and unauthenticated user.
- Apply the Principle of Least Privilege to all systems and services. Run all software as a non-privileged user (one without administrative privileges) to diminish the effects of a successful attack. (M1026: Privileged Account Management)
    - Safeguard 4.7: Manage Default Accounts on Enterprise Assets and Software: Manage default accounts on enterprise assets and software, such as root, administrator, and other pre-configured vendor accounts. Example implementations can include: disabling default accounts or making them unusable.
    - Safeguard 5.4: Restrict Administrator Privileges to Dedicated Administrator Accounts: Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.
    - Safeguard 5.5: Establish and Maintain an Inventory of Service Accounts: Establish and maintain an inventory of service accounts. The inventory, at a minimum, must contain department owner, review date, and purpose. Perform service account reviews to validate that all active accounts are authorized, on a recurring schedule at a minimum quarterly, or more frequently
- Architect sections of the network to isolate critical systems, functions, or resources. Use physical and logical segmentation to prevent access to potentially sensitive systems and information. Use a DMZ to contain any internet-facing services that should not be exposed from the internal network. Configure separate virtual private cloud (VPC) instances to isolate critical cloud systems. (M1030: Network Segmentation)
    - Safeguard 12.2: Establish and Maintain a Secure Network Architecture: Establish and maintain a secure network architecture. A secure network architecture must address segmentation, least privilege, and availability, at a minimum.
- Use capabilities to detect and block conditions that may lead to or be indicative of a software exploit occurring. (M1050: Exploit Protection)
    - Safeguard 10.5: Enable Anti-Exploitation Features: Enable anti-exploitation features on enterprise assets and software, where possible, such as Microsoft® Data

Execution Prevention (DEP), Windows® Defender Exploit Guard (WDEG), or Apple® System Integrity Protection (SIP) and Gatekeeper™.

- Restrict use of certain websites, block downloads/attachments, block Javascript, restrict browser extensions, etc. (M1021: Restrict Web-Based Content)
    - Safeguard 9.2: Use DNS Filtering Services: Use DNS filtering services on all enterprise assets to block access to known malicious domains.
    - Safeguard 9.3: Maintain and Enforce Network-Based URL Filters: Enforce and update network-based URL filters to limit an enterprise asset from connecting to potentially malicious or unapproved websites. Example implementations include category-based filtering, reputation-based filtering, or through the use of block lists. Enforce filters for all enterprise assets.
    - Safeguard 9.6: Block Unnecessary File Types: Block unnecessary file types attempting to enter the enterprise's email gateway.
- Remind users not to visit un-trusted websites or follow links provided by unknown or un-trusted sources. Inform and educate users regarding the threats posed by hypertext links contained in emails or attachments especially from un-trusted sources. (M1017: User Training)
    - Safeguard 14.1: Establish and Maintain a Security Awareness Program: Establish and maintain a security awareness program. The purpose of a security awareness program is to educate the enterprise's workforce on how to interact with enterprise assets and data in a secure manner. Conduct training at hire and, at a minimum, annually. Review and update content annually, or when significant enterprise changes occur that could impact this Safeguard.
    - Safeguard 14.2: Train Workforce Members to Recognize Social Engineering Attacks: Train workforce members to recognize social engineering attacks, such as phishing, pre-texting, and tailgating.

## References

https://www.jenkins.io/security/advisory/2026-09-16/

## CVEs

CVE-2026-92122
CVE-2026-92123
CVE-2026-92124
CVE-2026-92125
CVE-2026-92126
CVE-2026-92127
CVE-2026-92128
CVE-2026-92129
CVE-2026-92130
CVE-2026-92131
CVE-2026-92132
CVE-2026-92133
CVE-2026-92134
CVE-2026-92135
CVE-2026-92136
CVE-2026-92137
CVE-2026-92138
CVE-2026-92139
CVE-2026-92140
CVE-2026-92141

Smarttech
YOUR 24/7 SECURITY PARTNER

www.smarttech247.com