c3p0 prior to v0.12.0 can be dangerously abused to download and execute malicious code

High **swaldman** published **GHSA-5476-xc4j-rqcv** on Feb 23

Package

c3p0 (Java)

Affected versions	Patched versions
< 0.12.0	0.12.0+

c3p0-loom (Java)

< 0.12.0	0.12.0+

Description

Impact

c3p0 is vulnerable to attack via maliciously crafted Java-serialized objects and `javax.naming.Reference` instances. Several c3p0 `ConnectionPoolDataSource` implementations have a property called `userOverridesAsString` which conceptually represents a `Map<String,Map<String,String>>`. Prior to v0.12.0, that property was maintained as a hex-encoded serialized object. Any attacker able to reset this property, on an existing `ConnectionPoolDataSource` or via maliciously crafted serialized objects or `javax.naming.Reference` instances could be tailored execute unexpected code on the application's `CLASSPATH`.

The danger of this vulnerability was strongly magnified by vulnerabilities in c3p0's main dependency, mchange-commons-java. This library includes code that mirrors early implementations of JNDI functionality, including ungated support for remote `factoryClassLocation` values. Attackers could set c3p0's `userOverridesAsString` hex-encoded serialized objects that include objects "indirectly serialized" via JNDI references. Deserialization of those objects and dereferencing of the embedded `javax.naming.Reference` objects could provoke download and execution of malicious code from a remote `factoryClassLocation`.

Although hazard presented by c3p0's vulnerabilites are exarcerbated by vulnerabilities in mchange-commons-java, use of Java-serialized-object hex as the format for a writable Java-Bean property, of objects that may be exposed across JNDI interfaces, represents a serious independent fragility.

Patches

The `userOverridesAsString` property of c3p0 `ConnectionPoolDataSource` classes has been reimplemented to use a safe CSV-based format, rather than rely upon potentially dangerous Java object deserialization.

c3p0-0.12.0+ and above depend upon mchange-commons-java 0.4.0+, which gates support for remote `factoryClassLocation` values by configuration parameters that default to restrictive values. Those parameters are documented here.

c3p0 additionally enforces the new mchange-commons-java `com.mchange.v2.naming.nameGuardClassName` to prevent injection of unexpected, potentially remote JNDI names.

Workarounds

Users should upgrade to c3p0-0.12.0 or above. There is no supported workaround for earlier versions of c3p0.

References

c3p0, you little rascal — Hans-Martin Münch
c3p0 documentation, security note
c3p0 documentation, configuring security

Severity

High 8.9 / 10

CVSS v4 base metrics

Exploitability Metrics

Attack Vector	Adjacent
Attack Complexity	Low
Attack Requirements	Present
Privileges Required	Low
User interaction	None

Vulnerable System Impact Metrics

Confidentiality	High
Integrity	High

Availability	High
Subsequent System Impact Metrics	
Confidentiality	High
Integrity	High
Availability	High

Learn more about base metrics

CVSS:4.0/AV:A/AC:L/AT:P/PR:L/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H

CVE ID

CVE-2026-27830

Weaknesses

No CWEs

Credits

 **dpp** Reporter