2

2 vulnerabilities of arbitrary code in [redacted] - CVE-2017-5929

Share:

State	○ Resolved (Closed)
Disclosed	**October 9, 2019 12:10am +0530**
Reported To	**U.S. Dept Of Defense**
Weakness	Deserialization of Untrusted Data
Severity	Critical (9 ~ 10)
Participants	
Visibility	Disclosed (Full)

Collapse

TIMELINE · EXPORT

ruffdraft submitted a report to U.S. Dept Of Defense. Sep 29th (2 years ago)

Summary:

GitHub repo: https://github.com/%E2%96%88%E2%96%88%E2%96%88%E2%96%88%E2%96%88%E2%96%88%E2%96%88%E2%96%88

QOS.ch Logback before 1.2.0 has a serialization vulnerability affecting the SocketServer and ServerSocketReceiver components.

High Severity
Arbitrary Code Execution
Vulnerable module: ch.qos.logback:logback-core
Introduced through: com.github.dblock.waffle:waffle-distro@1.8.1
Detailed paths
Introduced through: [redacted]@[redacted]#a746bb4ecce1cb252a301c08be0daffa480c9747 › com.github.dblock.waffle:waffle-distro@1.8.1 › ch.qos.logback:logback-core@1.1.3
Introduced through: [redacted]@[redacted]#a746bb4ecce1cb252a301c08be0daffa480c9747 › com.github.dblock.waffle:waffle-distro@1.8.1 › ch.qos.logback:logback-classic@1.1.3 › ch.qos.logback:logback-core@1.1.3

and

High Severity
Arbitrary Code Execution
Vulnerable module: ch.qos.logback:logback-classic
Introduced through: com.github.dblock.waffle:waffle-distro@1.8.1
Detailed paths
Introduced through: [redacted]@[redacted]#a746bb4ecce1cb252a301c08be0daffa480c9747 › com.github.dblock.waffle:waffle-distro@1.8.1 › ch.qos.logback:logback-classic@1.1.3

Description:

ch.qos.logback:logback-core and ch.qos.logback:logback-classic Affected versions of this package are vulnerable Arbitrary Code Execution. A configuration can be turned on to allow remote logging through interfaces that accept untrusted serialized data. Authenticated attackers on the adjacent network can exploit this vulnerability to run arbitrary code through the deserialization of custom gadget chains.

Impact

Serialization is a process of converting an object into a sequence of bytes which can be persisted to a disk or database or can be sent through streams. The reverse process of creating object from sequence of bytes is called deserialization. Serialization is commonly used for communication (sharing objects between multiple hosts) and persistence (store the object state in a file or a database). It is an integral part of popular protocols like Remote Method Invocation (RMI), Java Management Extension (JMX), Java Messaging System (JMS), Action Message Format (AMF), Java Server Faces (JSF) ViewState, etc.
Deserialization of untrusted data (CWE-502), is when the application deserializes untrusted data without sufficiently verifying that the resulting data will be valid, letting the attacker to control the state or the flow of the execution.
Java deserialization issues have been known for years. However, interest in the issue intensified greatly in 2015, when classes that could be abused to achieve remote code execution were found in a popular library (Apache Commons Collection). These classes were used in zero-days

affecting IBM WebSphere, Oracle WebLogic and many other products.
An attacker just needs to identify a piece of software that has both a vulnerable class on its path, and performs deserialization on untrusted data. Then all they need to do is send the payload into the deserializer, getting the command executed.

Step-by-step Reproduction Instructions

1. Run known POC CVE online

Product, Version, and Configuration (If applicable)

ch.qos.logback:logback-core@1.1.3
ch.qos.logback:logback-classic@1.1.3

Suggested Mitigation/Remediation Actions

update to latest version

BOT: U.S. Dept Of Defense posted a comment. Sep 29th (2 years ago)

Greetings from the Department of Defense (DoD),

Thank you for supporting the DoD Vulnerability Disclosure Program (VDP).

By submitting this report, you acknowledge understanding of, and agreement to, the DoD Vulnerability Disclosure Policy as detailed at @DeptofDefense.

The VDP Team will review your report to ensure compliance with the DoD Vulnerability Disclosure Policy. If your report is determined to be out-of-scope, it will be closed without action.

We will attempt to validate in-scope vulnerability reports and may request additional information from you if necessary. We will forward reports with validated vulnerabilities to DoD system owners for their action.

Our goal is to provide you with status updates not less than every two weeks until the reported vulnerability is resolved.

Regards,

The VDP Team

opimentelsei updated the severity to Critical. Oct 2nd (2 years ago)

opimentelsei changed the status to ○ **Triaged**. Oct 2nd (2 years ago)

Greetings,

We have validated the vulnerability you reported and are preparing to forward this report to the affected DoD system owner for resolution.

Thank you for bringing this vulnerability to our attention!

We will endeavor to answer any questions the system owners may have regarding this report; however, there is a possibility we will need to contact you if they require more information to resolve the vulnerability.

You will receive another status update after we have confirmed your report has been resolved by the system owner. If you have any questions, please let me know.

Thanks again for supporting the DoD Vulnerability Disclosure Program.

Regards,

The VDP Team

ruffdraft posted a comment. Oct 2nd (2 years ago)

Thank you for the quick update

ruffdraft posted a comment. Oct 12th (2 years ago)

Hi. Any updates?

ruffdraft posted a comment. Updated Oct 4th (7 days ago)

Hi. I noticed a commit was added that fixed the issue. Will this be marked as resolved soon?
https://github.com/%E2%96%88%E2%96%88%E2%96%88%E2%96%88%E2%96%88%E2%96%88%E2%96%88%E2%96%88/commit/092e73ad15bbc98f195ca98ab8187641cf4da068I%E2%80%99ll

opimentelsei posted a comment. Updated Oct 4th (7 days ago)

@ruffdraft
Greetings,

We have been advised that the vulnerability you reported has been resolved. Before we close this report, we would appreciate it if you could confirm that the vulnerability no longer exists.

https://github.com/%E2%96%88%E2%96%88%E2%96%88%E2%96%88%E2%96%88%E2%96%88/pull/755

If we do not receive a response within two weeks, we will send you a second request. If we do not receive a response from you within two weeks of the second notice, we will close this report as resolved.

If you do not believe this vulnerability has been effectively resolved or if you have any questions, please let us know.

Thank you for your time and effort to improve the security of the DoD information network.

Regards,

The VDP Team

ruffdraft posted a comment. Oct 18th (2 years ago)

Yes. I looked and I do not see the vulnerability exists. Looks like it was fixed! Thanks!

opimentelsei closed the report and changed the status to ○ **Resolved**. Oct 18th (2 years ago)

Good news!

The vulnerability you reported is considered resolved and this report is now closed. If you have any questions, please let me know.

Thank you for your time and effort to improve the security of the DoD information network.

Regards,

The VDP Team

ruffdraft posted a comment. Oct 18th (2 years ago)

Thank you!

ruffdraft requested to disclose this report. Oct 18th (2 years ago)

agent-1 agreed to disclose this report. Oct 9th (3 days ago)

Disclosure approved

This report has been disclosed. Oct 9th (3 days ago)