CVE Response and FAQs

for Mirth® Connect by NextGen Healthcare, 4.4.1

NOTICE: This document contains information that is confidential and proprietary to NextGen Healthcare, Inc. and its subsidiaries and affiliates ("Company") and is intended for use solely by Company's authorized clients. This document may not be copied, reproduced, published, displayed, otherwise used, transmitted, or distributed in any form by any means as a whole or in any part, nor may any of the information it contains be used or stored in any information retrieval system or media, translated into another language, or otherwise made available or used by anyone other than the authorized client to whom this document was originally delivered without the prior, written consent of Company.

By retaining or using this document, you represent that you are a client or an authorized representative of a client of Company who is authorized to use this document under one or more agreements between you and Company now in force, and that you will use this document and the information it contains solely as and to the extent those agreements permit. Any other use or distribution of the contents of this document, as a whole or in any part, is prohibited. Although we exercised great care in creating this publication, Company assumes no responsibility for errors or omissions that may appear in this publication and reserves the right to change this publication at any time without notice.

Contents

Summary

An Unauthenticated Remote Command Execution vulnerability has been identified within Mirth Connect Core version 4.4.0 and lower. Below is a link to the independent vulnerability report that provides specific details of the vulnerability.

CVE-2023-43208 - NextGen Mirth Connect Remote Code Execution Vulnerability (CVE-2023-43208) – Horizon3.ai

Release notes for 4.4.1: 4.4.1 What's New · nextgenhealthcare/connect Wiki (github.com)

Impacted Versions

All versions of Mirth Connect 4.4.0 and earlier.

Mitigation

- Immediately ensure that your Administrator API port (default 8443) is not exposed to the public Internet. This limits the impact to your private network.
- Upgrade to Mirth Connect 4.4.1.

Resolution

The issue is resolved in Mirth Connect 4.4.1 and higher.

Frequently Asked Questions

Which versions of Mirth Connect are affected?

All Mirth Connect versions 4.4.0 and earlier.

Which Java version is affected?

All Java versions.

Is it just limited to the Mirth Connect Admin API port?

Yes, it is limited to the Administrator API port (by default 8443). It does not affect any channel interfaces like TCP Listener, HTTP Listener, etc.

What ports need to be blocked (i.e., to remove public access) if I do not upgrade to 4.4.1?

We encourage all clients to immediately upgrade to Mirth Connect 4.4.1. If you are unable to immediately upgrade, then the Admin API port (8443) should be blocked.

Does this affect the Web client default port (8080)?

No. The 8080 port is for the plain HTTP launch page, and for downloading the JNLP if needed. Both of those work fine on the HTTPS port too, so the HTTP port is not necessarily required. The plain HTTP port can be completely disabled by commenting out "http.port" in mirth.properties.

What has been done to confirm that no other components were impacted other than the Admin API?

The issue is related to deserialization done via a third-party library (Xstream). We reviewed other places in the Mirth Connect code base that uses XStream and determined that only the admin port uses the deserialization feature.

What new security testing was added to avoid introducing similar vulnerabilities in the future?

We have reviewed our unit tests and added new ones that will use Xstream. We use the best practice of testing in relation to the “allow list” instead of a “deny list”.

https://github.com/nextgenhealthcare/connect/commit/496357010cdd00bdd855d094b7153eae50e6bfeb

What documentation/certification exists regarding the vulnerability and how it was remediated?

The following links provide documentation that the CVEs fix has been corrected.

Independent assessment report:

CVE-2023-43208

- NextGen Mirth Connect Remote Code Execution Vulnerability (CVE-2023-43208) – Horizon3.ai

Release notes for version 4.4.1:

- 4.4.1 What's New · nextgenhealthcare/connect Wiki (github.com)

Has the patch been scanned for SAST, DAST, or SCA vulnerabilities? Is any internal security vulnerability testing conducted on Mirth Connect?

Yes, we have an Application Security Platform to scan our code base. SAST/DAST web inspect are done upon builds and we run a weekly third-party Dependency Checker – Open Source OWASP Tool.

Are there any processes for vulnerability resolution or mitigation for Mirth Connect?

Yes, we follow the NextGen Healthcare Critical Issue Management process that is maintained by our Quality Management Team.

What data or connections within Mirth Connect are encrypted or hashed? For these (if any), what ciphers or libraries are used to perform hashing or encryption?

Any connections using TLS (like HTTPS, TCPS, FTPS) or SSH (like SFTP) are encrypted. Message content can be encrypted in the database on a per-channel basis by enabling that in the channel settings. Passwords are salted/hashed in the database. As of version 4.3, encryption uses 128-bit AES/CBC/PKCS5Padding by default. As of version 4.4, hashing uses PBKDF2WithHmacSHA256 at 600000 iterations by default.

Where can I find the open-source pull request with changes to code in 4.4.1?

Below are the links to the GitHub pull request and specific sections of code that have been updated:

- https://github.com/nextgenhealthcare/connect/commit/b9e832d6e83ae6560fd2315a38d32787dddf58aa

- https://github.com/nextgenhealthcare/connect/blob/development/donkey/src/main/java/com/mirth/connect/donkey/util/xstream/XStreamSerializer.java#L67

Additional details are available about the mirth.properties settings when you search for "XStream" on the following page:

- https://docs.nextgen.com/bundle/Mirth_User_Guide_4_4_1/page/connect/connect/topics/c_The_mirth_properties_File_connect_ug.html

How do I upgrade from an older version of Mirth Connect to version 4.4.1?

Follow the standard upgrade process as described in Mirth Connect documentation available on GitHub.

- You can go directly from your current version to 4.4.1 but read below to determine if you want to upgrade in one step or multiple. Based on your custom setup, it is up to you to decide whether you upgrade directly or in smaller steps.
- Be sure to read the Release Notes (What’s New) and Upgrade Notes for every version you will upgrade to or through before upgrading.
- We recommend always upgrading in a development environment before moving to production.
- Open-source versions of Mirth Connect can be found at the NextGen Website as well as in GitHub and commercial extensions are available via our Success Community account.

Is there a plan to create patches for older versions?

No, not at this time. We encourage all customers to refer to the mitigation and upgrade suggestions documented above.

Who should I contact if I need help completing the upgrade?

Clients with a commercial Mirth Connect license can contact support through the NextGen Success Community: https://www.community.nextgen.com.

Open-source clients can access documentation, upgrade guides, and the online-support forums using the links below.

- Product documentation: https://docs.nextgen.com/category/mirth_connect
- Join the public Mirth Connect Slack Community: https://join.slack.com/t/mirthconnect/shared_invite/zt-1zf2dy4gh-cA~Xk~2tuF2T~Qj54aLK3Q
- Mirth Connect GitHub Discussions: https://github.com/nextgenhealthcare/connect/discussions
- Mirth Connect Forums: https://forums.mirthproject.io/index.php

Document Revision History

Date	Document Version	Summary of Changes
11/1/2023	2.0	Updated with minor edits to the Frequently Asked Questions section
11/1/2023	1.0	Initial release