

NICE CXone WebRTC Redirection

Receiver App Deployment Guide

Audience: Endpoint administrators, browser-policy owners

Status: Draft

Last updated: September 24, 2026

1. Purpose

This guide describes how to deploy and operate the **NICE CXone Receiver App** for **NICE WebRTC Redirection** on:

- Microsoft Windows 10/11
- Enterprise-managed Google ChromeOS
- Dell ThinOS 10.x (firmware 2508+)

With the Receiver App path:

- The CXone Agent UI remains in the hosted browser session.
- The NICE browser extension in that session is configured for the Receiver path.
- The **NICE CXone Receiver App** on the physical endpoint owns the microphone, speaker, and WebRTC media peer.
- The hosted session and the endpoint associate through the NICE matchmaking service; voice media then travels between the endpoint and NICE CXone.

For network readiness, STUN, and layered validation, see the [Customer Readiness and Validation Guide](#). For leadership checklist items, see the [Customer Readiness Executive Brief](#).

2. Architecture

CXone Agent UI + NICE browser extension → hosted browser session

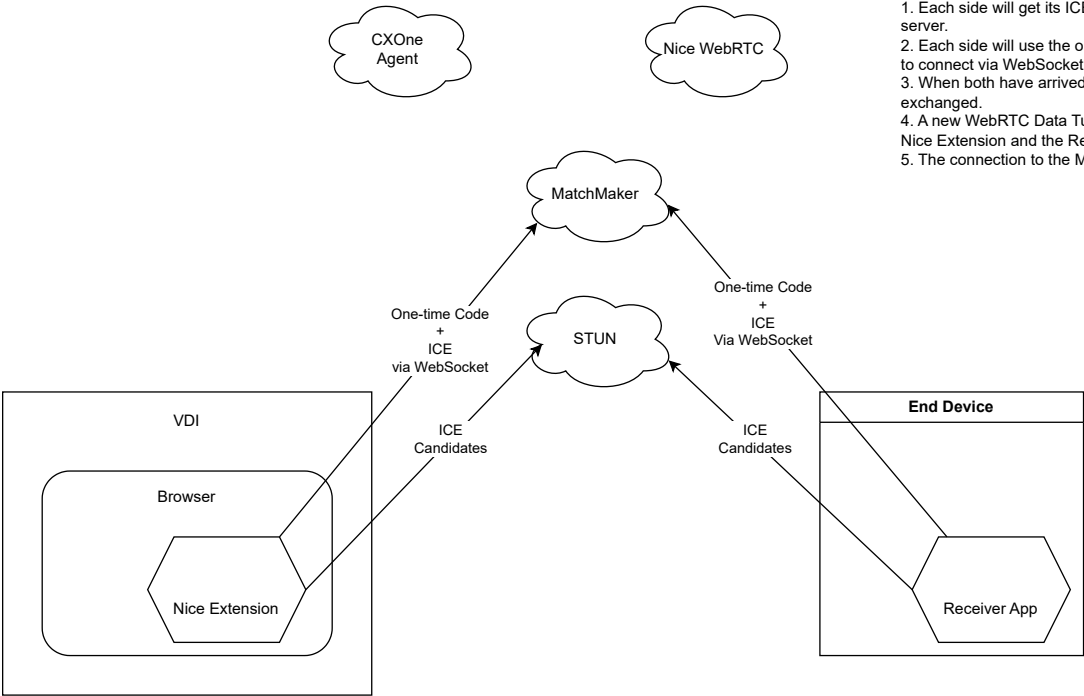
NICE matchmaking service (WSS, pairing only) → associates session ↔ endpoint

NICE CXone Receiver App → endpoint mic/speaker + RTP to CXone

Piece	Where it runs
CXone Agent UI	Browser in the hosted session
NICE browser extension	Same hosted browser
Matchmaking service	NICE (or customer-hosted) WSS endpoint — pairing and signaling handoff only ; not on the voice-media path
Microphone / speaker / RTP media	Physical endpoint via the NICE CXone Receiver App
NICE CXone WebRTC media	Direct from the endpoint (after pairing)

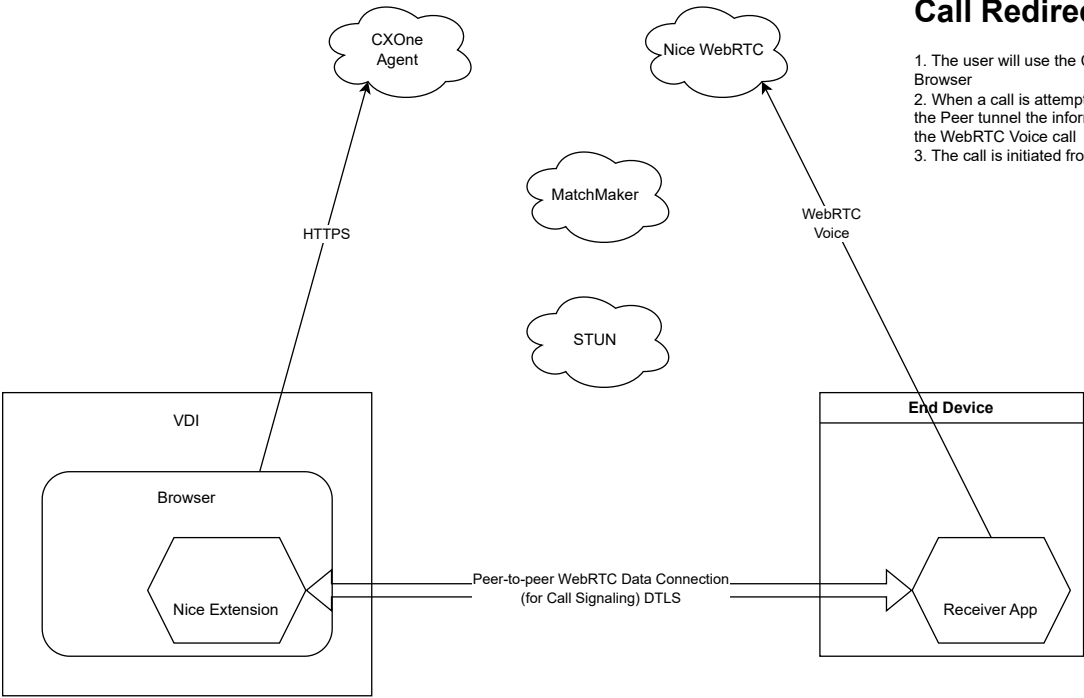
Matchmaking

1. Each side will get its ICE candidates from the STUN server.
2. Each side will use the one-time code (SPAKE 2 encrypted) to connect via WebSocket to the Matchmaker Service
3. When both have arrived, their ICE information is exchanged.
4. A new WebRTC Data Tunnel is established between the Nice Extension and the Receiver App
5. The connection to the Matchmaker service is then dropped



Call Redirection

1. The user will use the CXOne Agent web inside of the VDI Browser
2. When a call is attempted, the Nice Extension instead signals via the Peer tunnel the information that would be required to complete the WebRTC Voice call
3. The call is initiated from the Receiver App



Matchmaking

1. Each side obtains ICE candidates from the STUN service.
2. Each side uses the one-time code to connect over WebSocket to the matchmaking service.
3. When both have arrived, ICE information is exchanged.
4. A WebRTC data tunnel is established between the NICE browser extension and the Receiver.
5. The connection to the matchmaking service is then dropped.

Call redirection

1. The agent uses CXone Agent in the hosted browser.
2. When a call is attempted, the NICE extension signals the information required for the WebRTC voice call over the peer tunnel.
3. The call is initiated from the Receiver on the physical endpoint.

Important: A completed CXone call alone is **not** proof that redirection is active. Confirm pairing, matchmaker connectivity on both sides, and that media is owned by the **endpoint** Receiver (see [§10 Verification](#)).

3. Supported endpoints

Endpoint OS	Receiver form	Notes
Microsoft Windows 10/11 x64	Desktop app (MSI primary; NSIS secondary)	Reference platform for this path
Google ChromeOS (enterprise-managed)	Separate Manifest V3 Receiver extension on the Chromebook	Not an Electron app; requires Chrome Enterprise management
Dell ThinOS 10.x, firmware 2508+	Linux Receiver packaged via Dell App Builder → Wyse Management Suite	Documented; validate in your environment before production
Dell ThinOS 9.x	Not supported	No third-party application path

HP ThinPro, IGEL, and macOS endpoints are outside the scope of this guide.

ChromeOS without VDI: If CXone Agent runs natively in Chrome on the Chromebook (no hosted desktop session), WebRTC already uses on-device audio. Do **not** deploy the Receiver extension for that topology.

4. Version floors

Component	Required / recommended
NICE CXone Receiver App	2.x
Windows	Windows 10/11 x64
ChromeOS	Enterprise-managed Chromebook (Cloud Identity + ChromeOS Enterprise Upgrade or equivalent)
Dell ThinOS	ThinOS 10.x, build 2508 or later
Wyse Management Suite	5.3+ (ThinOS App Builder path)
Hosted browser	Google Chrome or Microsoft Edge (Chromium)
Hosted-session extension (enterprise)	oaaajlhoicikoedhcdcpaplhfjfmnkdlm
Hosted-session extension (Chrome Web Store)	hmfcdoeekoabipnapelhp1dlgioeab
ChromeOS Receiver (enterprise)	cclocilemnhgjmdklhooecnajedbgnf
ChromeOS Receiver (Chrome Web Store)	koidbbjfgcjclblphcbdooielhcadcap

Force-install **one** channel per browser profile. Block the other channel's IDs and the public NICE store ID `gcfbj1dfomnopndjajjfp1dkkdmmoi`.

Windows installers may be **unsigned** in early packages. Allowlist the EXE in endpoint security software and plan for Authenticode-signed builds from NICE before broad production rollout. ChromeOS production distribution uses the NICE-hosted update URL or the Chrome Web Store listing (see [ChromeOS Enterprise Deployment Guide](#)).

5. Shared policy and network

5.1 Hosted-session extension policy

Configure the NICE browser extension in the **hosted** Chrome or Edge session for the Receiver path.

Setting	Location	Required value
MediaProviderPreference	Extension managed policy (chrome.storage.managed)	MindfireRedirector
MatchmakerUrl	Extension managed policy	wss://liaison.nice-receiver-app.com/ws
PairingCodeEntry	Extension managed policy	UserPrompt , PolicySupplied , or Disabled
PairingCode	Extension managed policy	Full invitation when PairingCodeEntry is PolicySupplied
PairingCodeTtlMinutes	Extension managed policy	TTL for policy-supplied code (≥ 1)
AdditionalSupportedOrigins	Extension managed policy	Extra match patterns for custom CXone / Agent hosts, if needed
runtime_allowed_hosts	Chrome/Edge ExtensionSettings for the installed hosted-session ID	Same patterns as AdditionalSupportedOrigins (force-install does not auto-grant optional host permissions)
Receiver on endpoint	Per platform (\$6–\$8)	NICE CXone Receiver App (or ChromeOS Receiver extension) installed and able to reach the matchmaker

Important: The product name is **NICE CXone Receiver App**. The managed-policy value for this path is `MindfireRedirector` — that string is what administrators must set.

Example extension managed policy:

```
{
  "MediaProviderPreference": "MindfireRedirector",
  "MatchmakerUrl": "wss://liaison.nice-receiver-app.com/ws",
  "PairingCodeEntry": "UserPrompt",
  "DiagnosticsEnabled": true
}
```

Example force-install — **enterprise** channel:

```
{
  "oaaajlhoicikoedhcdcpaplhjfmnkdlm": {
    "installation_mode": "force_installed",
    "update_url": "https://extensions.nice-receiver-app.com/cxone-preview/updates.xml",
    "override_update_url": true
  },
  "hmfcadoeekoabipnapelhpdlgioeab": {
    "installation_mode": "blocked"
  },
  "gcfjbjldfomnopndpajjfpldkdkmmoi": {
    "installation_mode": "blocked"
  }
}
```

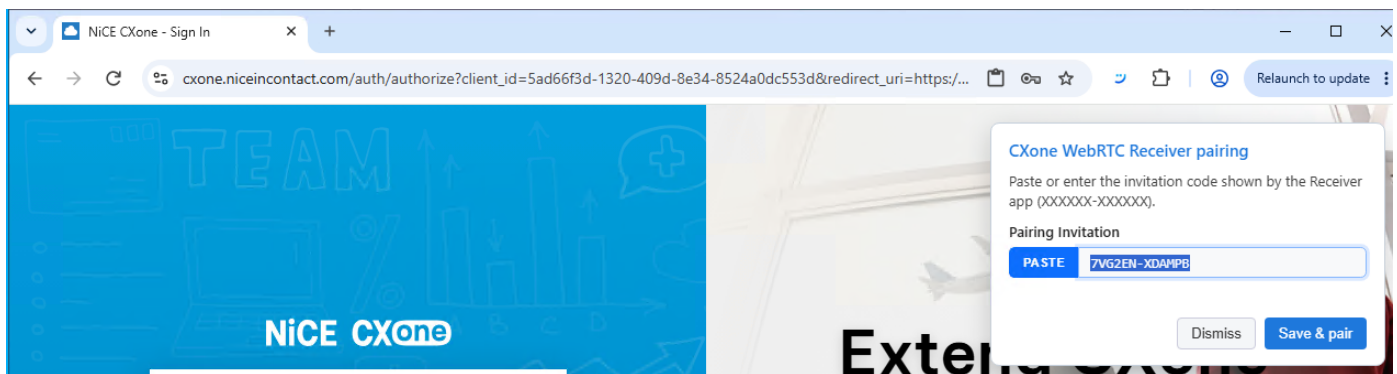
Example force-install — **Chrome Web Store** channel:

```
{
  "hmfcdoeekoabipnapelhpdlgioeab": {
    "installation_mode": "force_installed",
    "update_url": "https://clients2.google.com/service/update2/crx"
  },
  "oaaajlhoicikoedhcdcpaplhjfmnkdlm": {
    "installation_mode": "blocked"
  },
  "gcfjbjldfomnopnpdjajfpldkkdmmoi": {
    "installation_mode": "blocked"
  }
}
```

5.2 Pairing modes

Mode	Behavior	Typical use
UserPrompt	Agent enters the invitation from the Receiver UI into the extension	Pilot and day-to-day desks
PolicySupplied	Admin pushes the invitation via managed policy	Fixed desk / kiosk with a dedicated Receiver
Disabled	Receiver path unavailable in that session	Sessions that must not use Receiver

Invitation display format: XXXXXX-XXXXXX . After a successful pair, later calls reuse the persistent signaling channel; Reset in the Receiver UI (or a policy change) forces a new invitation.



5.3 Network

Open these paths from the networks agents actually use (office, home, VPN, branch). Hosted-desktop allowlists alone are not enough — voice leaves the **physical endpoint**.

Source	Destination (FQDN / IP)	Protocol / port
Hosted browser	liaison.nice-receiver-app.com / 135.237.20.87	TCP 443 (HTTPS/WSS)
Hosted browser	stun.nice-receiver-app.com (resolved A/AAAA)	UDP 19302
Physical endpoint	liaison.nice-receiver-app.com / 135.237.20.87	TCP 443 (HTTPS/WSS)
Physical endpoint	stun.nice-receiver-app.com (resolved A/AAAA)	UDP 19302
Physical endpoint	This tenant's NICE CXone cluster	TCP 443
Physical endpoint	This tenant's NICE CXone WebRTC infrastructure	UDP 1024–65535

Service	FQDN	IP address	Port
Matchmaking	liaison.nice-receiver-app.com	135.237.20.87	TCP 443 (wss://liaison.nice-receiver-app.com/ws)
STUN	stun.nice-receiver-app.com	DNS A/AAAA via CNAME to stun.l.google.com (rotates)	UDP 19302

Service	FQDN	IP address	Port
Documentation / downloads	docs.nice-receiver-app.com	Azure Static Web Apps A records (rotate)	TCP 443
Enterprise extension host	extensions.nice-receiver-app.com	Azure Static Web Apps A records (rotate)	TCP 443

There is **no TURN relay**. Prefer FQDN-based allowlisting for STUN and the extension host (destination IPs change). Use this tenant's NICE CXone Connectivity Requirements. TLS inspection and proxies must not break HTTPS/WSS to the matchmaking service.

6. Windows

6.1 Package and install

Item	Detail
Primary package	Per-machine MSI : NICE CXone Receiver App-Setup-{version}.msi
Secondary package	NSIS .exe (lab / legacy)
Install location	Program Files — executable name NICE CXone Receiver App.exe
Distribution	GPO, Intune, SCCM, or manual install for pilots
Auto-start	Common Startup logon shortcut (created by MSI and NSIS)
Firewall	Installer applies inbound/outbound TCP and UDP rules for the Receiver EXE

Upgrade: MSI→MSI upgrades share a common upgrade code. An older NSIS lab install is a different installer family — uninstall it before installing the MSI.

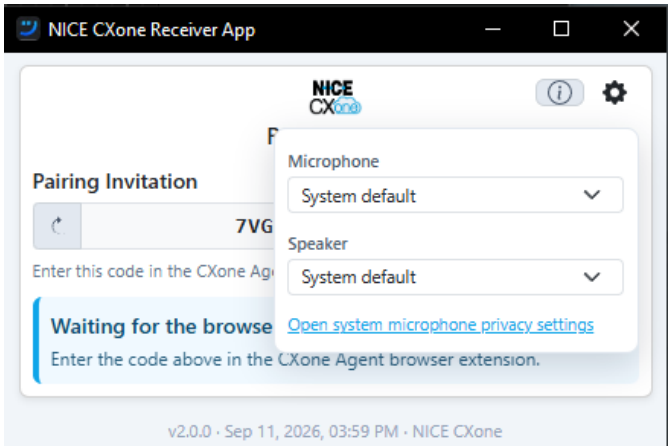
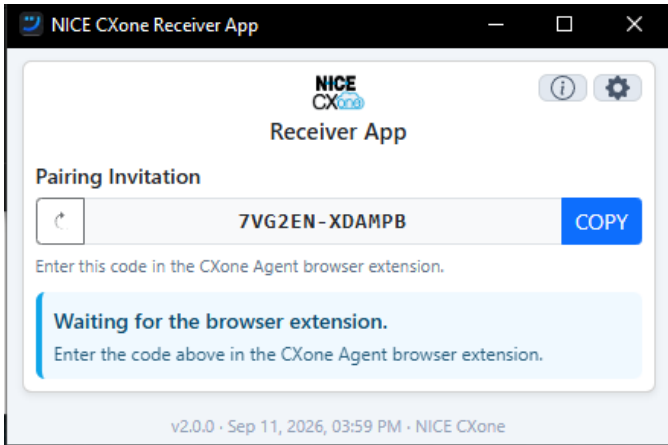
Trust / signing: Early Windows packages may be **unsigned**. Allowlist the EXE in endpoint security software and plan for Authenticode-signed builds from NICE before broad production rollout.

6.2 Endpoint requirements

- Windows Firewall (or equivalent) must allow the Receiver EXE for ICE/RTP (installer rules cover the default Windows Firewall case).
- Antivirus / EDR must permit **NICE CXone Receiver App.exe** to run and to access the microphone and speakers.
- Matchmaker WSS (TCP 443), STUN (UDP 19302), and CXone media UDP must be reachable from the endpoint (§5.3).

6.3 Operator surface

- While unpaired, the Receiver shows a bottom-right window (always-on-top) with the invitation and status.
- When paired, presence moves to the **system tray**; the window can be shown again from the tray.
- Header **Status** (info) exposes matchmaker URL, version, pairing state, and connection diagnostics when paired.



7. ChromeOS

ChromeOS cannot run the Windows/Linux Electron Receiver. The endpoint role is a **second, independently installed** Manifest V3 extension on the physical Chromebook — not the same package as the hosted-session NICE extension.

Google Admin click-path, force-install, and microphone pre-grant: [ChromeOS Enterprise Deployment Guide](#). Use that document for production rollout. The rest of this section is the short policy summary.

Piece	Where it runs	Role
NICE browser extension (hosted session)	Browser that hosts CXone (hosted desktop session)	Configured with <code>MediaProviderPreference = MindfireRedirector</code> ; pairs over the matchmaker
NICE CXone Receiver (ChromeOS)	Physical Chromebook only	Owens microphone, speaker, and the WebRTC media peer; toolbar popup is a remote control over an always-on offscreen document

Channel	Hosted-session ID	ChromeOS Receiver ID	Receiver update URL
Enterprise host	oaaajlhoicikoedhcdcpaplhjfmnkdlm	cclocilemnhgjmdklhoecnajedbgncf	<code>https://extensions.nice-receiver-app.com/chromeos-receiver/updates.xml</code>
Chrome Web Store	hmfcadoeeokoabipnapelhpdlgioeab	koidbbjfgcjclblphcbdooielhcadc	<code>https://clients2.google.com/service/update2/crx</code>

`AudioCaptureAllowedUrls` must include `chrome-extension://<RECEIVER_EXTENSION_ID>/` for the **installed** Receiver ID (trailing slash required).

7.1 Prerequisites

1. **Cloud Identity** (or Google Workspace) admin on a verified domain.
2. **ChromeOS Enterprise Upgrade** (or equivalent) licenses.
3. Device enrolled for enterprise management before user sign-in.

Consumer Google accounts alone are insufficient for force-install and microphone pre-grant.

7.2 Operator surface

The ChromeOS Receiver toolbar popup matches the Windows feature set for day-to-day use: invitation with Refresh/Copy, microphone grant and device pickers, Diagnostics, and Disconnect. Status uses the same five-state icon vocabulary as Windows (see [§9](#)).

8. Dell ThinOS 10

8.1 Requirements

- ThinOS **10.x**, firmware **2508+**
- Wyse Management Suite (**WMS**) **5.3+**
- **Enhanced Mode**, CI Software Enabler, and App Builder packages enabled per Dell guidance
- Customer CI signing certificate imported as `.crt` under **Advanced** → **Desktop Mode** → **CI Certificates** (not the general Certificates store)
- NICE-provided Linux Receiver package (`.deb` input to App Builder)

ThinOS 9.x is out of scope — there is no supported third-party application path.

8.2 Packaging and deployment

1. Obtain the NICE Linux Receiver `.deb` for the target release from NICE.
2. Wrap with Dell **App Builder** (`applicationType: CIApp` , record-0 name prefixed `ci_` , install timeout ≥ 600s).
3. Sign with the customer `.pem` ; import the matching `.crt` via CI Certificates.
4. Deploy through **Wyse Management Suite**.

8.3 WMS policy checklist

Setting	Required
Enhanced Mode + App Builder Mode	Enabled
CI Software Enabler	Installed
CI Certificate (<code>.crt</code>)	Imported under CI Certificates
<code>CIAppsEnableIcon</code>	Enabled
<code>CIAppAutoLaunchList</code>	Includes the Receiver app id
<code>CiApplications</code> Mic + Network capabilities	Granted (<code>EnhancedMode_Mic</code> , <code>EnhancedMode_Network</code>)
<code>AppEnvironmentVariables</code> / config file	<code>MATCHMAKER_URL=wss://...</code> (and optional <code>PROVISIONED_INVITATION</code> for fixed-desk pairing)
<code>AppLogPath</code>	<code>/var/log/nice-cxone-receiver</code> (collected via <code>/var/log/ci_<app></code>)

The Receiver runs as a Docker CI app (App Builder Isolated): process user `ubuntu` , Pulse audio via the launcher environment, host networking for ICE, no system tray (Weston task list), durable pairing under `/var/lib/nice-cxone-receiver` .

8.4 Customer-visible security tradeoff

Dell documents that **Enhanced Mode reduces overall system security**. Disclose this to security and desktop owners **before** enabling CI Receiver deployment. Prefer Dell-recommended **Isolated Mode** for CI apps where possible, and validate audio, UDP/ICE, and Operator UI behavior on your ThinOS image before production.

9. Operator pairing and status

Use this section to train agents and floor support. It is not a separate end-user booklet.

9.1 Pairing flow (UserPrompt)

1. Start the Receiver on the physical endpoint (or confirm the ChromeOS Receiver extension is loaded).
2. Copy the invitation (xxxxxx-xxxxxx) from the Receiver UI / popup.
3. In the hosted CXone session, enter the invitation where the NICE extension prompts for pairing (or use PolicySupplied invitations when configured).
4. Confirm the Receiver shows a connected / paired state before placing calls.
5. Use **Disconnect** / **Reset** only when intentionally ending the association (forces a new invitation).

9.2 Status icons

The same five-state vocabulary appears on Windows (tray / window) and ChromeOS (toolbar action):

State	Meaning (typical)
Gray	Idle / not ready
Blue	Connecting or transitional
Green	Ready / paired
On-call	Active redirected call
Error	Fault — check Diagnostics / Status and matchmaker connectivity

9.3 Diagnostics

Where available, Status / Diagnostics shows matchmaker URL and connection, signaling tunnel state, peer health, RTT, and ICE state. Collect these values with the Receiver version when opening a support case.

10. Verification

Do not treat a completed CXone call alone as proof of Receiver redirection. Confirm:

1. Receiver is running on the **physical endpoint** (Windows app, ThinOS package, or ChromeOS Receiver extension).
2. Hosted-session extension policy has `MediaProviderPreference = MindfireRedirector` and a reachable `MatchmakerUrl`.
3. Both the hosted browser and the endpoint can open WSS to the matchmaking service (TCP 443).
4. Pairing completes (Connected / paired in the Receiver UI).
5. During a call, audio is heard and spoken on **endpoint** devices, and Receiver Status / Diagnostics shows an active WebRTC session where available.

Check	Expected
Receiver running	Version visible; available for pairing
Matchmaker	Connected from hosted session and endpoint
Pairing	Invitation accepted; persistent association retained across calls until Reset

Check	Expected
Media	Endpoint mic/speaker; not “silent success” with media stuck in the hosted session only

11. Failure symptoms

Symptom	Likely cause
Receiver never shows a usable invitation	App/extension not installed or not started; ChromeOS Receiver ID / force-install missing
Pairing never completes	Matchmaker blocked (TCP 443 / TLS inspection); wrong <code>MatchmakerUrl</code> ; invitation expired or mistyped
Extension does not offer Receiver pairing	<code>MediaProviderPreference</code> not set to <code>MindfireRedirector</code>
Mic denied / no devices	Windows privacy or EDR blocking the EXE; ChromeOS missing <code>AudioCaptureAllowedUrls</code> or Grant microphone not completed
Call “works” but audio is poor / not on headset	Media not on the endpoint path — re-check pairing and Receiver session state
ThinOS package will not install or run	Firmware below 2508; Enhanced Mode / CI / App Builder packages missing; package not signed for WMS
Cannot paste invitation into hosted session (ChromeOS)	Clipboard policy between Chromebook and hosted desktop session blocked
ChromeOS unmanaged device	Enterprise enrollment required for production force-install and mic pre-grant

12. Production checklist

Shared

- ☐ Hosted Chrome/Edge: NICE extension force-installed (**one** channel ID from §4)
- ☐ Competing channel IDs and public store ID `gcfjb1ldfomnopndjajjfp1dkkdmmoi` blocked
- ☐ Extension managed policy: `MediaProviderPreference` = `MindfireRedirector`
- ☐ Extension managed policy: `MatchmakerUrl` = `wss://liaison.nice-receiver-app.com/ws`
- ☐ Pairing mode chosen (`UserPrompt` or `PolicySupplied`) and documented for agents
- ☐ Network: matchmaker `liaison.nice-receiver-app.com` / `135.237.20.87` TCP 443 from hosted browser **and** endpoint; STUN `stun.nice-receiver-app.com` UDP 19302 from both; CXone media UDP/HTTPS per tenant requirements
- ☐ Validation: pairing Connected; media on endpoint; Receiver Diagnostics available to support

Windows

- ☐ MSI (preferred) deployed via GPO / Intune / SCCM (or signed package when NICE provides it)
- ☐ Common Startup shortcut present; Receiver starts at logon
- ☐ Firewall rules applied for `NICE CXone Receiver App.exe`
- ☐ AV/EDR allowlist permits run + microphone/speaker access
- ☐ NSIS lab builds uninstalled before MSI production install where applicable

ChromeOS

- ☐ Devices enterprise-enrolled with ChromeOS Enterprise Upgrade (or equivalent)
- ☐ Receiver extension force-installed by **Receiver** ID for the chosen channel
- ☐ `AudioCaptureAllowedUrls` includes `chrome-extension://<RECEIVER_EXTENSION_ID>/`
- ☐ Clipboard allows pairing-code paste into hosted session when required
- ☐ Native-only CXone-on-Chrome topologies identified so Receiver is **not** forced there unnecessarily

Dell ThinOS 10

- ☐ Firmware **2508+**; WMS **5.3+**

- ☐ Enhanced Mode + CI Software Enabler + App Builder approved with security disclosure
 - ☐ CI certificate imported as `.crt` under Advanced → Desktop Mode → CI Certificates
 - ☐ `CIAppsEnableIcon` + `CIAppAutoLaunchList` configured
 - ☐ `CiApplications` grants Mic + Network; matchmaker URL via env/config
 - ☐ App Builder package signed and assigned in WMS
 - ☐ Isolated Mode validated for Pulse socket audio, UDP/ICE, Weston task-list UI
 - ☐ ThinOS 9.x endpoints excluded from this path
-
-

13. Q&A

Business leaders

What platforms does the Receiver App cover?

Microsoft Windows 10/11, enterprise-managed ChromeOS, and Dell ThinOS 10.x (firmware 2508+). HP ThinPro, IGEL, and macOS are outside this product documentation set.

Do agents need training beyond pairing?

For `UserPrompt` pairing, agents enter or paste a one-time code. For `PolicySupplied` pairing, association can be driven by policy with less agent interaction. Day-to-day CXone Agent usage stays the same.

IT administrators

MSI or EXE for Windows?

Prefer the per-machine MSI for production (GPO/Intune/SCCM). Use the NSIS EXE for lab or legacy only; uninstall lab builds before MSI production install.

Why is the ChromeOS Receiver a different extension ID?

ChromeOS cannot run the desktop Receiver EXE. The Chromebook runs a Manifest V3 Receiver extension force-installed by Google Admin. The hosted session still runs the hosted-session extension with `MindfireRedirector`.

What breaks microphone on ChromeOS most often?

Force-install alone does not grant capture. `AudioCaptureAllowedUrls` must include `chrome-extension://<RECEIVER_EXTENSION_ID>/` (trailing slash), and audio policy must not disable the Receiver.

ThinOS — what is non-negotiable?

Firmware 2508+, Enhanced Mode / CI / App Builder packages, Mic + Network grants, and Isolated Mode validation for Pulse audio and UDP/ICE. ThinOS 9.x is not this path.

VDI administrators

When should we use Receiver instead of Citrix UCSDK or Horizon?

Use Receiver when the endpoint is Windows/ChromeOS/ThinOS without (or outside) vendor WebRTC offload for CXone, or when you standardize on NICE pairing across mixed endpoint fleets.

Does the hosted browser still need the NICE extension?

Yes. Set `MediaProviderPreference` to `MindfireRedirector` and `MatchmakerUrl` to `wss://liaison.nice-receiver-app.com/ws`.

Can Receiver and UCSDK run together?

No. Choose one connection style per session. Do not force `MindfireRedirector` on a Citrix UCSDK or Horizon WebRTC path.

Network administrators

What must the physical endpoint reach?

Matchmaker `liaison.nice-receiver-app.com` / 135.237.20.87 (TCP 443), STUN `stun.nice-receiver-app.com` (UDP 19302), and this tenant's CXone HTTPS/UDP media paths. The hosted browser needs matchmaker and STUN as well.

Does ThinOS Isolated Mode change firewall rules?

Destination requirements are the same; confirm UDP/ICE and matchmaker WSS still work under Isolated Mode on a pilot device before wide rollout.

14. References

NICE product documentation

- [Customer Readiness Executive Brief](#) — leadership checklist
- [Customer Readiness and Validation Guide](#) — network, STUN, layered validation
- [ChromeOS Enterprise Deployment Guide](#) — Google Admin force-install and microphone pre-grant
- [Citrix UCSDK Policy Guide](#) — alternate connection style
- [Omnissa Horizon Policy Guide](#) — alternate connection style