

NICE CXone WebRTC Redirection

ChromeOS Enterprise Deployment Guide

Audience: Google Workspace / Chrome Enterprise administrators, endpoint owners

Status: Draft

Last updated: September 24, 2026

1. Purpose

This guide describes how to install and operate the **NICE CXone Receiver** on enterprise-managed Chromebooks using the **Google Admin console** (Google’s Chrome Enterprise management).

ChromeOS cannot run the Windows/Linux Receiver desktop app. The endpoint role is a **Manifest V3 Chrome extension** force-installed onto the physical Chromebook. After pairing, that extension owns the microphone, speaker, and WebRTC media peer.

This document also covers **microphone access**, which is the most common ChromeOS failure mode. Force-installing the extension does **not** grant the microphone. Capture happens in a hidden offscreen document that Chrome cannot prompt from, so the fleet **must** pre-grant the extension origin in Admin policy.

For network readiness (STUN, CXone media ports), see the [Customer Readiness and Validation Guide](#). For the Receiver path on Windows and ThinOS, see the [Receiver App Deployment Guide](#).

2. What you are installing

Two different Chrome extensions exist. They have different IDs, different update URLs, and different microphone policies. Do not reuse one ID for the other.

Piece	Where it runs	Role
NICE CXone Receiver (this guide)	Physical Chromebook	Owens microphone, speaker, and RTP media. Toolbar popup is a remote control over an always-on offscreen document.
NICE browser extension (hosted session)	Browser that hosts CXone Agent (hosted desktop, or Chrome on the Chromebook)	Redirector. Must be set to <code>MediaProviderPreference = MindfireRedirector</code> so it pairs with the Receiver over the matchmaker. That string is the required managed-policy value for the Receiver path.
NICE matchmaking service	NICE (or customer-hosted) WSS	Associates the hosted session with the Chromebook. Not on the voice-media path.

CXone Agent UI + NICE browser extension → hosted browser session

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

NICE CXone Receiver extension → Chromebook mic/speaker + RTP to CXone

Do not deploy the Receiver when CXone Agent runs natively in Chrome on the Chromebook with no hosted desktop session. Native WebRTC already uses on-device audio.

3. Prerequisites

Requirement	Why it is required
Google Admin console on a verified domain (Google Workspace or Cloud Identity)	Only a managed tenant can force-install extensions and pre-grant the microphone.
ChromeOS Enterprise Upgrade (or equivalent) licenses assigned to the Chromebooks	Consumer Chromebooks and unmanaged accounts cannot receive <code>ExtensionInstallForcelist</code> or <code>AudioCaptureAllowedUrls</code> .
Devices enterprise-enrolled (<code>Ctrl+Alt+E</code> at the sign-in screen, before a user signs in)	Policy applies to enrolled ChromeOS devices. Powerwash first if a consumer user already signed in.
Organizational unit (OU) for agent Chromebooks / agent users	Apply force-install and microphone policy to a pilot OU first, then widen.
Chromebooks can reach the matchmaker on TCP 443 (<code>wss://liaison.nice-receiver-app.com/ws</code> , IP <code>135.237.20.87</code>), STUN on UDP 19302 (<code>stun.nice-receiver-app.com</code>), and this tenant's CXone media paths	Voice leaves the physical Chromebook, not the hosted desktop.

A personal Gmail account, developer mode, or "Load unpacked" is **not** a production install path.

4. Identifiers (use these exact values)

Choose **one** distribution channel per organizational unit. Force-install **only one channel** per profile — never both.

Item	Enterprise host	Chrome Web Store
Receiver extension ID	<code>cclocilemnhgjmdklhooecnajedbgncf</code>	<code>koidbbjfgcjclblphcbdooielhcadcap</code>
Receiver update URL	<code>https://extensions.nice-receiver-app.com/chromeos-receiver/updates.xml</code>	<code>https://clients2.google.com/service/update2/crx</code>
Microphone origin	<code>chrome-extension://cclocilemnhgjmdklhooecnajedbgncf/</code>	<code>chrome-extension://koidbbjfgcjclblphcbdooielhcadcap/</code>
Hosted-session (VDI) extension ID	<code>oaaajlhoicikoedhcdcpaplhjfmnkdlm</code>	<code>hmfcadoeekoabipnapelhpdlgioeab</code>
Hosted-session update URL	<code>https://extensions.nice-receiver-app.com/cxone-preview/updates.xml</code>	<code>https://clients2.google.com/service/update2/crx</code>

Also **block** the public NICE store ID `gcfjbjldfomnopndjajjfp1dkkdmmoi` on managed browsers, and block the channel IDs you are **not** installing.

The trailing slash on the microphone origin is required. Copy the origin exactly as shown. Do **not** put the hosted-session ID or the NICE public store ID into Receiver force-install or microphone policy.

5. Force-install the Receiver in Google Admin

These steps use Google's current Chrome Enterprise navigation. Labels can move slightly between Admin console refreshes; search for the setting name if the menu path differs.

The steps below use the **enterprise host** IDs. For a Chrome Web Store OU, substitute the Chrome Web Store Receiver ID and update URL from §4, and block the enterprise IDs instead.

5.1 Open Apps & extensions for the pilot OU

1. Sign in to the [Google Admin console](#) with a Super Admin or Chrome administrator account.
2. Go to **Devices** → **Chrome** → **Apps & extensions**.
 - If the tenant uses Chrome Enterprise Core / Browser Cloud Management only, the equivalent path is **Chrome browser** → **Apps & extensions**.
3. At the top, select **Users & browsers**.
 - Use this tab for agents who sign in with a managed Google account on enrolled Chromebooks.
 - If agents use **managed guest sessions**, repeat the same add-and-force-install steps on the **Managed guest sessions** tab. Policy does not copy automatically between tabs.
4. On the left, select the **organizational unit** (or group) that contains the agent users. Leave the top OU selected only if every user in the tenant should receive the Receiver.

5.2 Add the extension by ID from the custom update URL

The Receiver is **not** the public Chrome Web Store "CXone Agent WebRTC Extension". Add it by ID from NICE's update service (enterprise channel) or from the Chrome Web Store (store channel).

1. Click **Add** (yellow +).
2. Choose **Add Chrome app or extension by ID**.
3. Enter the Receiver ID for your channel (enterprise example):

```
cclocilemnhgjmdklhooecnajedbgncf
```

4. For the enterprise channel, select **From a custom URL**. For the Chrome Web Store channel, use the store lookup / store update URL instead.
5. Enterprise channel — enter the update manifest URL:

```
https://extensions.nice-receiver-app.com/chromeos-receiver/updates.xml
```

The URL must be `https://`. Chrome will not force-install from `http://`.

6. Click **Save**.

If you skip **From a custom URL** on the enterprise channel, Admin looks the ID up in the Chrome Web Store. That lookup fails (or installs the wrong product) because this ID is the enterprise-signed Receiver, not a public listing.

5.3 Set installation policy

1. Click the **NICE CXone Receiver** row you just added.
2. In the right-hand panel, under **Installation policy**, choose:

Policy	When to use it
Force install + pin to ChromeOS taskbar	Required for production. Installs silently, users cannot remove it, and the icon stays on the shelf so agents can open the pairing popup.
Force install	Acceptable only if you pin the icon another way. Agents still need a reliable way to open the popup.
Allow install / Block	Do not use. Agents will not get a consistent install, and microphone policy becomes a per-user lottery.

3. Under deployment / update settings (wording varies by console version):
 - **Installation URL** — the same update URL from §5.2 for your channel.
 - **Update URL** — for the enterprise channel, choose **Installation URL** (this is `override_update_url`). If you leave updates on the value inside the CRX manifest, Chrome may stop pulling NICE-hosted builds.
4. Click **Save**.

Force-installed apps still download even when the Chrome Web Store is turned off for the OU.

5.4 Equivalent JSON — enterprise OU (optional)

If your tenant uses the **ExtensionSettings** JSON editor instead of the Apps & extensions UI, this is the same policy:

```
{
  "cclocilemnhgjmdklhooecnajedbgnfc": {
    "installation_mode": "force_installed",
    "update_url": "https://extensions.nice-receiver-app.com/chromeos-receiver/updates.xml",
    "override_update_url": true,
    "toolbar_pin": "force_pinned"
  },
  "oaaajlhoicikoedhcdcpaplhjfmnkdlm": {
    "installation_mode": "force_installed",
    "update_url": "https://extensions.nice-receiver-app.com/cxone-preview/updates.xml",
    "override_update_url": true
  },
  "gcfjbjldfomnopnnpjajjfpldkkdmmoi": {
    "installation_mode": "blocked"
  },
  "koidbbjfgcjclblphcbdooielhcadcap": {
    "installation_mode": "blocked"
  },
}
```

```

    "hmfcdoeekoabipnapelhpdlgioeab": {
      "installation_mode": "blocked"
    }
  }
}

```

`force_installed` and `normal_installed` both require `update_url`. You cannot use `"*"` as the ID for force-install.

5.5 Equivalent JSON — Chrome Web Store OU (optional)

For an OU that must use the Chrome Web Store listings instead of the enterprise update host. **Block** the enterprise IDs and the NICE public listing:

```

{
  "koidbbjfgcjclblphcbdooielhcadcap": {
    "installation_mode": "force_installed",
    "update_url": "https://clients2.google.com/service/update2/crx",
    "toolbar_pin": "force_pinned"
  },
  "hmfcdoeekoabipnapelhpdlgioeab": {
    "installation_mode": "force_installed",
    "update_url": "https://clients2.google.com/service/update2/crx"
  },
  "cclocilemnhgjmdklhooecnajedbgnfc": {
    "installation_mode": "blocked"
  },
  "oaaajlhoicikoedhcdcpaplhjfmnkdlm": {
    "installation_mode": "blocked"
  },
  "gcfjbjldfomnopnpdjajjfpldkkdmmoi": {
    "installation_mode": "blocked"
  }
}

```

Microphone allowlist for a Chrome Web Store OU must use the **Chrome Web Store** Receiver origin, not the enterprise one:

```
chrome-extension://koidbbjfgcjclblphcbdooielhcadcap/
```

Optional hardening: list **both** Receiver origins under Audio input allowed URLs so switching channels does not require a second mic-policy edit. Still force-install only one Receiver ID.

6. Microphone permission (required — do not skip)

This section is the ChromeOS-specific work. A Windows MSI can request the microphone as a desktop app. A ChromeOS **plug-in cannot**. If this policy is missing, calls often **connect and then go silent**, and the Receiver hangs up after about 30 seconds.

6.1 Why a ChromeOS plug-in cannot “just ask”

The Receiver captures audio in a Manifest V3 **offscreen document**. That page is hidden. Chrome only shows a microphone prompt on a **visible** surface (a tab or the toolbar popup).

Consequences:

1. When a call starts, the offscreen document calls `getUserMedia`. If permission is still **Ask**, Chrome does **not** show a dialog. Capture fails.
2. The first failed attempt is often stored as **Block** for the extension origin. After that, Chrome will never prompt again.
3. Force-install grants some enterprise APIs (for example device attributes). It does **not** grant `getUserMedia`.
4. Allowlisting CXone **websites** (`https://*.niceincontact.com`, Salesforce, and so on) does nothing for the Receiver. Capture is requested by the extension origin, not by the CXone tab.

The supported fleet fix is Google Admin **Audio input allowed URLs** (`AudioCaptureAllowedUrls`). That policy grants the origin without a prompt, so the offscreen document can capture on the first call.

6.2 Three layers that can all block the mic

All three must allow capture. Fixing only one is not enough.

Layer	Where it lives	What “good” looks like
1. ChromeOS hardware / org master switch	Admin: Devices → Chrome → Settings → Users & browsers → Hardware → Audio input (microphone)	Audio input is enabled (or disabled only if the Receiver origin is still on the allowlist in layer 2).
2. Extension origin allowlist (required)	Same Hardware section: Audio input allowed URLs	Exactly the origin for the installed Receiver ID (see §4), including the trailing slash
3. Device Privacy controls	On the Chromebook: Settings → Privacy and security → Privacy controls → Microphone	Microphone access is on, and Chrome / the Receiver is not denied at OS level.

Layer 2 is the one administrators miss. Layer 3 is the one floor support hits when a single agent toggled Privacy controls off.

6.3 Set Audio input allowed URLs in Google Admin

1. In Admin console go to **Devices** → **Chrome** → **Settings**.
2. Open **Users & browsers** (this page opens by default).
3. Select the **same OU** you used for force-install.
4. Scroll to **Hardware** (or search for `Audio input`).
5. Configure:

Audio input (microphone)

Leave this **enabled** so websites and apps may use the microphone.

If your security baseline **disables** audio input for the OU, capture is then allowed **only** for URLs on the allowlist below — which is acceptable **only if** the Receiver origin is on that list. Disabling audio input **and** omitting the Receiver origin blocks every call.

Audio input allowed URLs (`AudioCaptureAllowedUrls`)

Add this **one** origin for the enterprise channel, on its own line (substitute the Chrome Web Store origin if that is your channel):

```
chrome-extension://cclocilemnhgjmdklhooecnajedbgnf/
```

6. Click **Save**.

Do not add CXone site URLs here as a substitute for the extension origin. You may add CXone sites as well if other tools need them; they do not replace the Receiver entry.

6.4 URL format rules (this policy is strict)

Entry	Result
<code>chrome-extension://cclocilemnhgjmdklhooecnajedbgnf/</code>	Correct for the enterprise Receiver. Trailing slash included.
<code>chrome-extension://koidbbjfgcjclblphcbdooielhcadcap/</code>	Correct for the Chrome Web Store Receiver.
<code>chrome-extension://cclocilemnhgjmdklhooecnajedbgnf</code>	Often does not match . Always include the trailing slash.
<code>https://chrome-extension://...</code>	Invalid.
<code>*</code> or <code><all_urls></code>	Not supported by this policy. Chrome ignores it.
<code>chrome-extension://oaaajlhoicikoedhcdcpaplhjfmnkdlm/</code>	Wrong product (hosted-session extension).
<code>chrome-extension://gcfjbjldfomnopnjdajjfpldkkdmmoi/</code>	Wrong product (public store CXone listing).
<code>https://*.niceincontact.com</code>	Grants CXone pages , not the Receiver plug-in.

Chrome matches the **security origin** of the requester, not a file path inside the extension. The origin is `chrome-extension://<id>/`.

6.5 What agents should see after policy applies

On a correctly managed Chromebook, agents should **not** have to click **Grant microphone access**. Device names appear in the Receiver popup pickers, and the first call captures audio.

If a pilot device still prompts or shows “Microphone access is blocked”:

1. Confirm `AudioCaptureAllowedUrls` on `chrome://policy` (see §9). Reboot the Chromebook once after the first policy push — ChromeOS often applies extension and hardware policy at startup.
2. Confirm Privacy controls (layer 3) are not off.
3. Only then use the popup fallback in §6.6.

6.6 Fallback: Grant microphone access in the popup (lab / one-off)

Use this when policy is not in place yet (lab) or a single profile stored a Block.

1. Click the **NICE CXone Receiver** icon on the ChromeOS shelf (this is why §5.3 pins it).
2. Open the gear / settings control in the popup.
3. Click **Grant microphone access**.

4. Allow the Chrome prompt if it appears.

That click runs `getUserMedia` in the **visible popup**. Once the `chrome-extension://` origin is granted, the offscreen document can capture for the rest of the profile without prompting again.

If the button immediately shows **Microphone access is blocked** and no prompt appears, Chrome already stored Block. Reset it:

1. In the same popup, click **Reset microphone permission in Chrome settings** → (opens the site-details page for this extension).
2. Set **Microphone** from **Block** to **Ask**, or remove the override.
3. Alternatively open `chrome://settings/content/microphone`, find the Receiver origin under **Not allowed to use your microphone**, and delete it.
4. Return to the popup and click **Grant microphone access** again. Allow the prompt.
5. If the OS Privacy controls page is what blocked it, turn **Microphone access** on under **Settings** → **Privacy and security** → **Privacy controls**.

There is no API the extension can call to clear a stored Block. Policy (`AudioCaptureAllowedUrls`) is the only way to keep a fleet out of this state.

6.7 Policies that accidentally undo the grant

Admin setting	Effect
Audio input (microphone) disabled, and Receiver origin not on Audio input allowed URLs	No capture, including the Receiver.
<code>ExtensionSettings</code> blocked_permissions including audio-related permissions	Extension may install but cannot capture. Do not block audio for this ID.
A Block these sites / content-settings list that includes the Receiver origin	Overrides the allowlist. Remove the Receiver origin from any block list.
Applying microphone policy to a different OU than force-install	Extension installs; mic policy never reaches those users. Use the same OU (or a parent that both inherit).

Video / camera allowlists are **not** required for this product.

7. Hosted-session extension and pairing policy

The Chromebook Receiver only helps if the **hosted** CXone browser is on the Receiver path.

On the Chrome/Edge that runs CXone Agent (VDA image, or Chrome on the Chromebook for HTML5 Workspace):

Setting	Value
Force-install ID	Match the channel from §4 (<code>oaaajlhoicikoedhcdcpaplhjfmnkdlm</code> enterprise, or <code>hmfcdoeekoabipnapelhpdlgioeab</code> Chrome Web Store)
Update URL	Matching URL from §4
Block public store listing	<code>gcfjbjldfomnopndjajjfpldkkdmmoi</code> → <code>installation_mode: blocked</code>

Setting	Value
Managed policy MediaProviderPreference	MindfireRedirector
Managed policy MatchmakerUrl	wss://liaison.nice-receiver-app.com/ws
Managed policy PairingCodeEntry	UserPrompt for pilots

Do **not** set MindfireRedirector on sessions that should use Citrix UCSDK or Omnissa Horizon instead. Those paths never pair with the ChromeOS Receiver.

Details for the hosted-session extension are in the [Receiver App Deployment Guide](#) §5.

7.1 Clipboard into a hosted desktop (Citrix / HTML5 on ChromeOS)

If CXone runs inside Citrix Workspace (or similar) on the Chromebook, the agent copies the pairing code from the Receiver popup and pastes it into the hosted session. Allow clipboard from ChromeOS Chrome into that session. A blocked clipboard looks like “Receiver is fine, hosted session never gets the code.”

8. Enroll a Chromebook (if it is not already managed)

1. On a new device, complete Wi-Fi setup. **Do not sign in** a user yet.
2. At the sign-in screen press **Ctrl+Alt+E**.
3. Sign in with a Google Admin account for the customer domain and complete enrollment.
4. Confirm a ChromeOS Enterprise Upgrade license is assigned to the device.
5. Move the device and/or user into the pilot OU from §5.

If someone already signed in with a consumer account, **Powerwash** first, then enroll. Policy will not attach to an unmanaged consumer profile.

9. Verify on a Chromebook

After saving Admin policy, wait a few minutes, then **reboot the Chromebook once**.

9.1 chrome://policy

On the Chromebook, open chrome://policy → **Reload policies** → enable **Show policies with no value set**.

Policy	Expected
ExtensionSettings or ExtensionInstallForcelist	Receiver ID for your channel, force_installed , matching update URL
AudioCaptureAllowedUrIs	Contains the microphone origin for the installed Receiver ID
AudioCaptureAllowed	Enabled / unset (not forced off unless the allowlist is also present)

Status for those rows should be **OK**, not Conflict / Ignored / Error. Click **Show more** on ExtensionSettings and confirm the JSON is valid.

9.2 chrome://extensions

Check	Expected
NICE CXone Receiver present	Enabled, version matches the hosted package
ID on the card	The Receiver ID for your channel from §4
Remove	Not available (force-installed)
Errors on the card	None

9.3 Receiver popup

Click the shelf icon.

Check	Expected
Status	Waiting with a XXXXXX-XXXXXX invitation, not a red pairing error
Microphone	Device names in the pickers without clicking Grant (if §6 applied)
Diagnostics → Matchmaker URL	Production <code>wss://liaison.nice-receiver-app.com/ws</code> (or the URL this build was packaged with)

A completed CXone call with audio on the **hosted desktop** is **not** success. Confirm the Receiver Diagnostics row **Call streaming = Yes** and that the Chromebook, not the VDA, owns the microphone.

10. Failure symptoms

Symptom	Likely cause
Extension never appears	Wrong OU; device not enrolled; custom URL omitted so Admin searched the Web Store; Chromebook cannot fetch <code>updates.xml</code> (TLS inspection / firewall)
Wrong extension installed	Used the public store ID <code>gcfjbjldfomnopndjajjfp1dkkdmmoi</code> or the hosted-session ID
Users can remove the extension	Installation policy is Allow, not Force install
Pairing code never appears / red banner	Matchmaker unreachable from the Chromebook (TCP 443 / TLS); wrong baked-in matchmaker URL
Call connects, no audio, hangs up ~30s	Microphone not granted to the extension origin. Set Audio input allowed URLs; see §6
Grant microphone access does nothing / "blocked"	Stored Block on the origin, or Privacy controls off. Reset per §6.6
Mic works in Meet / CXone website but not in Receiver	Site allowlists are not the extension origin. Add the Receiver <code>chrome-extension://.../</code> origin
Hosted session never offers Receiver pairing	Hosted extension <code>MediaProviderPreference</code> is not <code>MindfireRedirector</code>
Cannot paste invitation into Citrix	Clipboard between ChromeOS and the hosted session blocked

Symptom	Likely cause
Policy missing on <code>chrome://policy</code>	Applied to a different OU; user not in that OU; device needs reboot

11. Production checklist

- ☐ Pilot OU selected; Chromebooks enrolled with ChromeOS Enterprise Upgrade
- ☐ Distribution channel chosen: **enterprise host** or **Chrome Web Store** (not both on the same OU)
- ☐ Receiver added **by ID** with the matching update URL
- ☐ Installation policy **Force install + pin to ChromeOS taskbar**
- ☐ Update URL overridden to the Installation URL (enterprise channel)
- ☐ Competing channel IDs **blocked**; NICE store ID `gcfjbjldfomnopndjajjfp1dkkdmmoi` blocked
- ☐ **Audio input (microphone)** not disabling the Receiver
- ☐ **Audio input allowed URLs** contains the origin for the **installed** Receiver ID (trailing slash); optionally both origins when dual-channel is in play
- ☐ Same OU used for force-install and microphone policy
- ☐ No block-list / `blocked_permissions` undoing audio for this ID
- ☐ Hosted-session extension on the Receiver path (`MindfireRedirector` + same matchmaker)
- ☐ Clipboard allowed into the hosted session when CXone is inside Workspace / HTML5
- ☐ `chrome://policy` on a pilot device shows force-install + `AudioCaptureAllowedUrls` = OK
- ☐ Pilot: pair, inbound call, outbound call, mute, hold; **Call streaming = Yes** on the Chromebook
- ☐ Pilot: reboot Chromebook; Receiver returns on startup
- ☐ Support knows §6.6 reset steps and how to open Receiver Status / Diagnostics

12. Q&A

Business leaders

Why is ChromeOS a separate guide?

Chromebooks cannot run the Windows/Linux Receiver desktop app. Production requires enterprise enrollment, ChromeOS Enterprise Upgrade (or equivalent), and Google Admin force-install plus microphone pre-grant.

Can agents use personal Chromebooks?

Not for production. Unmanaged / consumer accounts cannot receive force-install or reliable microphone pre-grant policy.

IT administrators

Force-install succeeded — why is there still no microphone?

Add the Receiver origin to **Audio input allowed URLs** (`chrome-extension://<ID>/` with trailing slash) on the same OU. Confirm Hardware → Audio input is not disabling capture for that app.

Enterprise host or Chrome Web Store?

Pick one channel per OU. Never force-install both on the same profile. Block the channel you are not using and block the public NICE store listing.

Where do I verify policy actually applied?

On a pilot Chromebook open `chrome://policy` and confirm force-install and `AudioCaptureAllowedUrls` show OK for the installed Receiver ID.

Do we still install an extension in the hosted session?

Yes — the hosted-session extension on the Receiver path (`MindfireRedirector` + same matchmaker). The ChromeOS Receiver extension is only on the physical Chromebook.

VDI administrators

We use Citrix Workspace HTML5 on Chromebooks — anything special?

Clipboard between Chromebook and hosted session must allow pasting the pairing code when agents use invitation pairing. Media still leaves the Chromebook Receiver, not the hosted desktop.

Should we force-install the Receiver on devices that only run native CXone in Chrome?

Identify native-only topologies so the Receiver is not forced where pairing is never used.

Network administrators

What must Chromebooks reach?

Matchmaker `wss://liaison.nice-receiver-app.com/ws` (135.237.20.87, TCP 443), STUN `stun.nice-receiver-app.com` (UDP 19302), enterprise updates from `extensions.nice-receiver-app.com` if using that channel, and this tenant's CXone media paths.

Is STUN required on the Chromebook itself?

Yes. The ChromeOS Receiver gathers ICE candidates on the physical device; block STUN on the Chromebook network and redirection will not establish cleanly.

13. References

Google

- [Automatically install apps and extensions](#) — Force install in Admin console
- [View and configure apps and extensions](#) — Add by ID, custom URL
- [Configure ExtensionSettings policy](#) — `force_installed`, `update_url`, `override_update_url`
- [Set Chrome policies for users or browsers](#) — Hardware, Audio input, Audio input allowed URLs
- [AudioCaptureAllowedUrls](#) — Policy definition
- [Manage camera, microphone & location on a Chromebook](#) — Device Privacy controls

NICE

- [Receiver App Deployment Guide](#) — Windows, ChromeOS summary, ThinOS
- [Customer Readiness and Validation Guide](#) — Network, STUN, validation

- [Customer Readiness Executive Brief](#) — Leadership checklist