Environments where CRYPE communication have only internet access through a proxy server

CRYPE communication requires some specific adjustments and configuration to work. Below is an explanation of how we make CRYPE function in this context.

1. Challenges with CRYPE in Proxy Environments

1. **CRYPE Peer-to-Peer (P2P) Communication**: CRYPE attempts to establish direct P2P connections, which often fail when proxies block UDP traffic or require HTTP/HTTPS-only communication.

2. **STUN/TURN Server Communication**: STUN servers may not be reachable if UDP is blocked. TURN servers, however, can relay traffic over TCP/TLS, which is often allowed by proxies.

3. **CRYPE Apps**: Unlike browsers, which respect system-level proxy configurations automatically, CRYPEapps require additional configuration to route traffic through the proxy.

2. Solutions for CRYPE Over Proxies

a. Use TURN Servers for Relayed Communication

When direct P2P communication is blocked due to the proxy:

- TURN servers can relay CRYPE traffic, bypassing the need for direct UDP connections.
- TURN servers should be configured to use **TCP** and **TLS**:
- TCP ensures that communication works even when UDP is blocked.
- TLS can bypass restrictive HTTPS-only proxies.

b. Configuration of CRYPE App with Proxies

Desktop

CRYPE applications don’t automatically respect system-level proxy configurations for non-HTTP(S) traffic, such as CRYPE Signaling or TURN server connections. You need to configure Apps networking explicitly.

1. **Proxy Configuration**: Use session.setProxy()API to set a proxy for all network requests:
2. **App with TURN/STUN**: Ensure the proxy configuration allows connections to the TURN server over TCP/TLS:

- Whitelist TURN server domains in the proxy.
- Ensure the TURN server supports transport over TCP or TLS.

Mobile

CRYPE mobile apps are designed to run on mobile devices or as a hybrid app. To support CRYPE with proxies:

1. **System Proxy on Mobile**:
   - Mobile devices usually respect system-level proxy configurations automatically.
   - Ensure the proxy allows CRYPE signaling (to the signaling server) and TURN server connections.

2. **Custom Proxy Configuration**:
   - Use community/http to configure proxy-aware HTTP/TLS traffic for CRYPE signaling or TURN communication.
   - Alternatively, create a custom networking layer to route all CRYPE-related traffic through the proxy.

c. CRYPE Signaling Through Proxy

Your signaling server, which coordinates CRYPE connections by exchanging communication candidates, must also work through the proxy.

1. **WebSocket or HTTPS for Signaling**:
   - Ensure that the CRYPE signaling server communicates over WebSocket (wss://) or HTTPS (https://).
   - Most proxies allow these protocols without additional configuration.

2. **Proxy Configuration for Signaling**:For Desktop, you can explicitly route signaling traffic through the proxy
   - For mobile, use the native HTTP/WebSocket library plugin to ensure signaling traffic respects the proxy.

3. Practical Considerations

1. **TURN Server Dependency**:
   - CRYPE in proxy environments almost always requires a TURN server, as direct P2P connections are unlikely to succeed.
   - Ensure our TURN server supports TCP and TLS.

2. **Proxy Whitelisting**:
   - Ask the enterprise IT team to whitelist our TURN server's IP/domain and ports.
   - Ensure the signaling server (WebSocket/HTTPS) is also accessible through the proxy.