**#16064** -<https://redmine.vnc.biz/issues/16064>

1. **Description**:

Steps to reproduce:

1)Ravi invites Dhaval for video chat

2)Dhaval accepts invitation.

3)Ravi starts screensharing session

4)Initially Dhaval can see Ravi's shared screen

5)Ravi stops screensharing

6)After sometime Ravi starts screensharing again,dhaval cant see shared screen or Dhaval cant see video of Ravi as well.

1. **Analytic**
   1. **Root cause:**

Can’t install directly chrome extension from IFrame.

Function checkExtInstalled is out updated.

* 1. **Solution:**

Handle install by message from jitsiMeet to JappixMini

Update some function in mini.js.

1. **Implement**

|  |
| --- |
| Enable desktop sharing in jitsiMett config.js  desktopSharing: 'ext', |
| Update function to check is installed VNCtalk Desktop Streamer in desktopsharing.js  function checkExtInstalled(callback) {  if (!chrome.runtime) {  // No API, so no extension for sure  callback(false, false);  return;  }  chrome.runtime.sendMessage(  config.chromeExtensionId,  { getVersion: true },  function (response) {  if (!response || !response.version) {  // Communication failure - assume that no endpoint exists  console.warn(  "Extension not installed?: ", chrome.runtime.lastError);  callback(false, false);  return;  }  // Check installed extension version  var extVersion = response.version;  console.log('Extension version is: ' + extVersion);  var updateRequired = isUpdateRequired(config.minChromeExtVersion, extVersion);  callback(!updateRequired, updateRequired);  }  );  }    Send a message with extension ID to jappixMini:  function obtainScreenFromExtension(streamCallback, failCallback) {  checkExtInstalled(  function (isInstalled) {  if (isInstalled) {  doGetStreamFromExtension(streamCallback, failCallback);  } else {  //Send message to required install VNC extension  switchInProgress = false;  connection.emuc.sendMessage(config.chromeExtensionId, "requiredInstallExtension");  }  }  );  } |
| Drop this message in chat room, edit muc.js  onMessage: function (msg) {  .....  if(nick == "requiredInstallExtension") {  //message for installing required extension for desktop sharing  return true;  }  ....  } |
| Handle message in jappixMini and dialog message for user, edit mini.js  //Handle install required extension for screen sharing  if(jQuery(node).find('>nick[xmlns="http://jabber.org/protocol/nick"]').text() == "requiredInstallExtension") {  if (!biz\_zcs\_vnc\_talk\_zimlet\_HandlerObject.isConferenceRunning()) {  return;  }  var nick\_name = from.split("/")[1].replace("\_video","");  var current\_name = con.jid.split('@')[0];  if(nick\_name == current\_name) {    var dialog\_install = appCtxt.getYesNoMsgDialog();// get a simple message dialog  dialog\_install.reset(); // reset the dialog  dialog\_install.setMessage("You must install VNC Desktop Streamer to use this feature. Click 'Yes' to install.", DwtMessageDialog.INFO\_STYLE);    dialog\_install.setButtonListener(DwtDialog.YES\_BUTTON,new AjxListener(this,function(){  dialog\_install.popdown();  self.initInlineInstalls(body);  self.installRequiredExtension(body);    }));  dialog\_install.setButtonListener(DwtDialog.NO\_BUTTON,new AjxListener(this,function(){  dialog\_install.popdown();  console.log("CANCEL suggest installing VNC Desktop Streamer.");    }));  dialog\_install.popup();  return;  }  }    Create some new function in mini.js  var isInstalled = false;  //Check extension installed  self.checkExtensionInstalled = function(ext\_id){  if (!chrome.runtime) {  return false;  }  chrome.runtime.sendMessage(  ext\_id,  { getVersion: true },  function (response) {  if (!response || !response.version) {  isInstalled = false;  }else{  isInstalled = true;  // Check installed extension version  var extVersion = response.version;  console.log('Extension version is: ' + extVersion);  }  }  );  };  self.initInlineInstalls = function(ext\_id){  var url = "https://chrome.google.com/webstore/detail/" + ext\_id;  var link = '<link rel="chrome-webstore-item" href="' + url + '">';  var check\_link = jQuery('link[rel="chrome-webstore-item"]');  if (check\_link.length == 0) {  jQuery('head').append(link);  }else{  jQuery("link[rel=chrome-webstore-item]").attr("href", url);  }  };  self.OpenInNewTab = function(url){  var win = window.open(url, '\_blank');  win.focus();  };  //Install extension for screen sharing feature  self.installRequiredExtension = function(ext\_id){  if (!chrome.runtime) {  return;  }  self.checkExtensionInstalled(ext\_id);  var url = "https://chrome.google.com/webstore/detail/" + ext\_id;  if (!isInstalled) {  chrome.webstore.install(  url,  function (arg) {  console.log("Extension installed successfully", arg);  var dialog\_success = appCtxt.getYesNoMsgDialog();// get a simple message dialog  dialog\_success.reset(); // reset the dialog  dialog\_success.setMessage("You must reload your browser to use new extension. Click 'Yes' to reload now, 'No' to reload later.", DwtMessageDialog.INFO\_STYLE);  dialog\_success.setButtonListener(DwtDialog.YES\_BUTTON,new AjxListener(this,function(){  dialog\_success.popdown();  // We need to reload the page in order to get the access to  window.location.reload(false);  }));  dialog\_success.setButtonListener(DwtDialog.NO\_BUTTON,new AjxListener(this,function(){  dialog\_success.popdown();  console.log("Reload browser later.");  }));  dialog\_success.popup();    },  function (arg) {  console.error("Failed to install the extension", arg);  var dialog\_fail = appCtxt.getYesNoMsgDialog();// get a simple message dialog  dialog\_fail.reset(); // reset the dialog  dialog\_fail.setMessage("Auto installing VNC Desktop Stream extension was failed, Please install it manually in chrome web store. Click 'Yes' to install now, 'No' to install later.", DwtMessageDialog.INFO\_STYLE);  dialog\_fail.setButtonListener(DwtDialog.YES\_BUTTON,new AjxListener(this,function(){  dialog\_fail.popdown();  self.OpenInNewTab(url);  }));  dialog\_fail.setButtonListener(DwtDialog.NO\_BUTTON,new AjxListener(this,function(){  dialog\_fail.popdown();  console.log("Install extension later.");  }));  dialog\_fail.popup();  }  );  }  }; |