1. Abkbmnbcmfehcbfjkpagdmloiondlbne TP
from cs_window_eventListener_message to chrome_tabs_executeScript_sink
cs.js

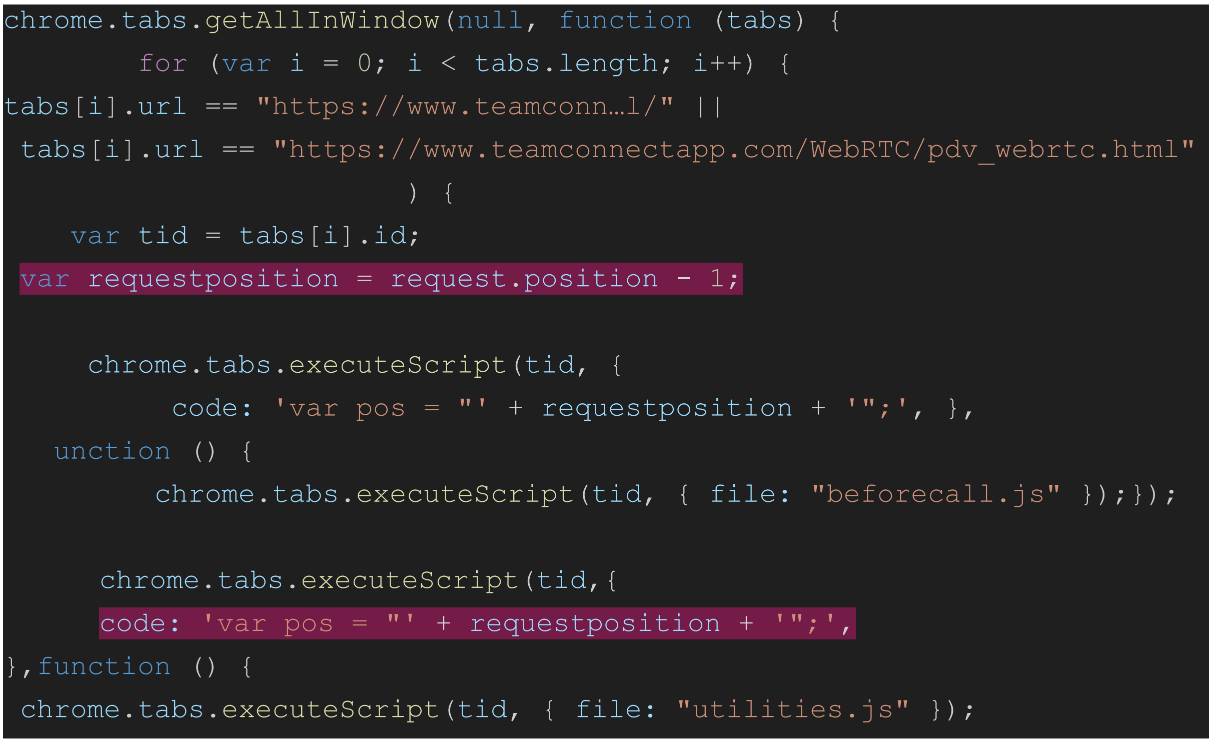
```
window.addEventListener("message", function(event) {
    if (event.source != window)
        return;

    if (event.data.type && (event.data.type == "LV_PCIDSS_TRIGGER")) {
        console.log("LiquidPause.Event: " + event.data.event);
        chrome.runtime.sendMessage(event.data);
    }
});
```

bg.js

```

chrome.runtime.onMessage.addListener(handleEvent);
function handleEvent(request, sender, sendResponse) {
  chrome.tabs.query({'active': true, 'windowId':
chrome.windows.WINDOW_ID_CURRENT},
           function(tab){
              var curTitle = tab[0].title;
              console.log('LiquidPause.Background.Process: ' +
request.event)
              clearTimeout(delayResume);
              if (request.event == "PAUSE" &&
curTitle.endsWith(request.windowTitle) == false)
              {

chrome.tabs.executeScript(tab.id,{code:"document.title = '" + curTitle
+ request.windowTitle + "'"});
              }
```

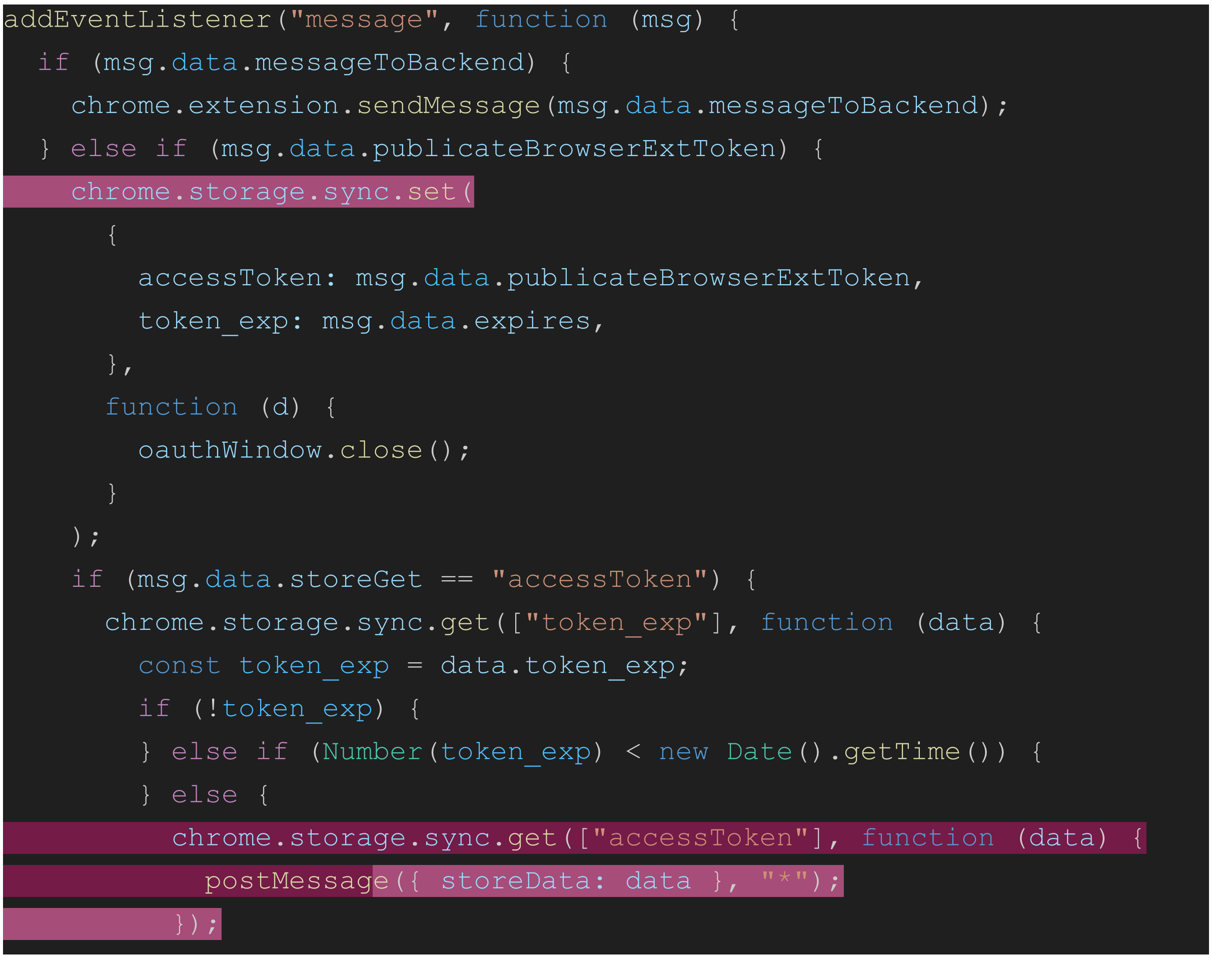
2. Acijcblmeekcgkheikedbjjgigkeefkf   TP
Sink: bg_chrome_runtime_MessageExternal → chrome_tabs_executeScript_sink
Execute code with attacker controlled data
bg.js

```
chrome.runtime.onMessageExternal.addListener(function (request, sender,
sendResponse) {
if (request.type == "notificationMouseDown"){
chrome.tabs.query({url:
"https://www.teamconnectapp.com/WebRTC/pdv_webrtc.html/*"},
function(results) {
      if (results.length == 0) {
….
 }else{
```

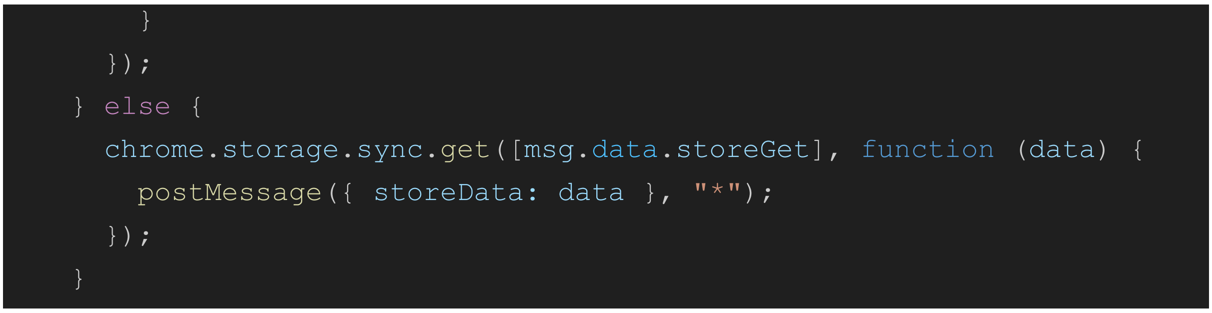


```
chrome.tabs.getAllInWindow(null, function (tabs) {
        for (var i = 0; i < tabs.length; i++) {
tabs[i].url == "https://www.teamconn…l/" ||
 tabs[i].url == "https://www.teamconnectapp.com/WebRTC/pdv_webrtc.html"
                        ) {
    var tid = tabs[i].id;
 var requestposition = request.position - 1;

     chrome.tabs.executeScript(tid, {
          code: 'var pos = "' + requestposition + '";', },
   unction () {
         chrome.tabs.executeScript(tid, { file: "beforecall.js" });});

      chrome.tabs.executeScript(tid,{
      code: 'var pos = "' + requestposition + '";',
},function () {
 chrome.tabs.executeScript(tid, { file: "utilities.js" });
```

3. Adahoneonjbcodnbkdngadoffhdekhnf TP
Sink: cs_window_eventListener_message → chrome_storage_sync_set_sink
Set and GET

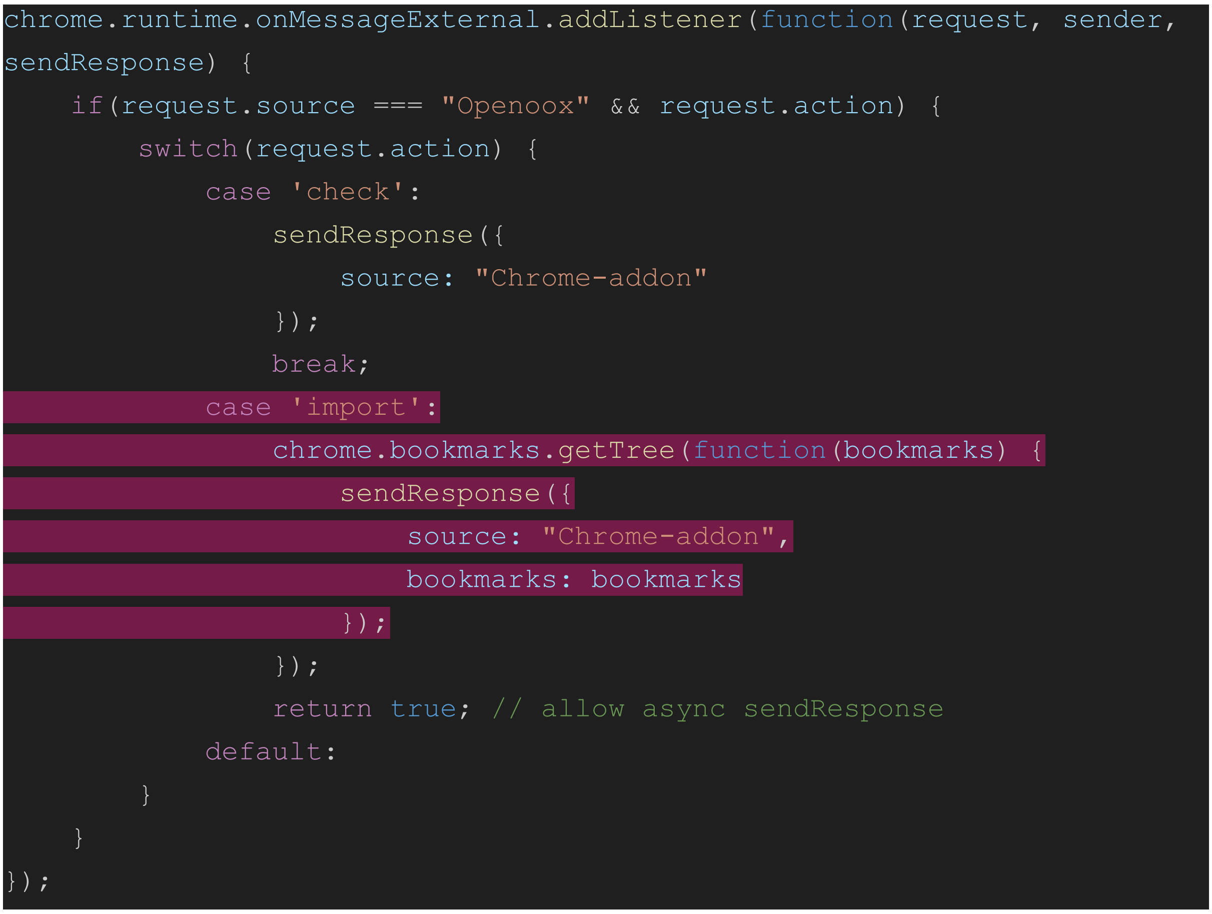


```
addEventListener("message", function (msg) {
  if (msg.data.messageToBackend) {
    chrome.extension.sendMessage(msg.data.messageToBackend);
  } else if (msg.data.publicateBrowserExtToken) {
    chrome.storage.sync.set(
      {
        accessToken: msg.data.publicateBrowserExtToken,
        token_exp: msg.data.expires,
      },
      function (d) {
        oauthWindow.close();
      }
    );
    if (msg.data.storeGet == "accessToken") {
      chrome.storage.sync.get(["token_exp"], function (data) {
        const token_exp = data.token_exp;
        if (!token_exp) {
        } else if (Number(token_exp) < new Date().getTime()) {
        } else {
          chrome.storage.sync.get(["accessToken"], function (data) {
            postMessage({ storeData: data }, "*");
          });
```



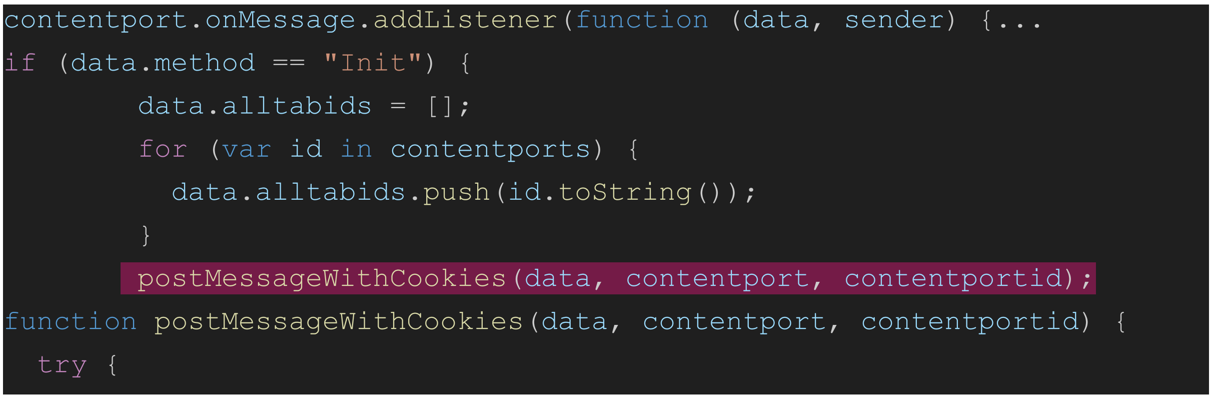
```
        }
      });
    } else {
      chrome.storage.sync.get([msg.data.storeGet], function (data) {
        postMessage({ storeData: data }, "*");
      });
    }
```

4. Aefmgkhgcmdljpfijlohmbhkhflmbmfi TP
from BookmarkTreeNode_source to sendResponseExternal_sink



```
chrome.runtime.onMessageExternal.addListener(function(request, sender,
sendResponse) {
    if(request.source === "Openoox" && request.action) {
        switch(request.action) {
            case 'check':
                sendResponse({
                    source: "Chrome-addon"
                });
                break;
            case 'import':
                chrome.bookmarks.getTree(function(bookmarks) {
                    sendResponse({
                        source: "Chrome-addon",
                        bookmarks: bookmarks
                    });
                });
                return true; // allow async sendResponse
            default:
        }
    }
});
```

5. Aicohdnbjhigmdoeilbigahinnmcnnnk TP
from cookies_source to externalNativePortpostMessage_sink
Get cookies and send back



```
contentport.onMessage.addListener(function (data, sender) {...
if (data.method == "Init") {
        data.alltabids = [];
        for (var id in contentports) {
          data.alltabids.push(id.toString());
        }
        postMessageWithCookies(data, contentport, contentportid);
function postMessageWithCookies(data, contentport, contentportid) {
  try {
    chrome.cookies.getAllCookieStores(function (stores) {
      try {
        var storeid = null;
        for (var i = 0; i < stores.length; i++) {
          for (var j = 0; j < stores[i].tabIds.length; j++) {
            if (stores[i].tabIds[j] === contentportid) {
              storeid = stores[i].id;
              break;
            }
          }
          if (storeid) {
            break;
          }
        }
        chrome.cookies.getAll({ url: data.srcurl, storeId: storeid },
function (cookies) {
          try {
            var cookiestr = "";
            if (cookies) {
              for (var i = 0; i < cookies.length; i++) {
                var cookie = cookies[i];
                cookiestr += (cookiestr ? "; " : "") + cookie.name +
"=" + cookie.value;
              }
            }
            data.indata.cookies = cookiestr;
          } catch (e) {
            handleError(e);
          }
          try {
            port.postMessage(data);
```

6. ajlbdflhaaflcepndpkdgejimggjcpnm FP
Hardcoded url
from cookies_source to externalNativePortpostMessage_sink
from cookies_source to window_postMessage_sink

```
var port = chrome.runtime.connectNative("com.fabasoft.nmhostpm17");
```

cookiestr += (cookiestr ? "; " : "") + cookie.name + "=" + cookie.value
Cs script send to bg where cookies are get and posted back

```
if (!alreadyactive) window.addEventListener("message", function(event)
{
  if (event.source !== window) {
    return;
  }
  if (windoworigin && event.origin !== windoworigin) {
    return;
  }
  // --
  if (event.data.type && (event.data.type ==
"com.fabasoft.nm.sendpm17")) {
    if (!windoworigin) {
      windoworigin = event.source.origin;
    }...
          if (windoworigin) {
            window.postMessage(response, windoworigin);
          }
```

Bg script

```
chrome.runtime.onConnect.addListener(function(contentport) {
  var contentportid = contentport.sender.tab.id + "#" +
nextcontentportid++;
  var released = false;
  contentports[contentportid] = contentport;
  contentport.onMessage.addListener(function(data, sender) {
...
    try {
      data.srcid = contentportid.toString();
      if (data.method == "Init") {
        data.alltabids = [];
        for (var id in contentports) {
          data.alltabids.push(id.toString());
        }
        postMessageWithCookies(data, contentport, contentportid);
      }
      else if (data.method == "UpdateLoginToken") {
        postMessageWithCookies(data, contentport, contentportid);
      }
      else {
        port.postMessage(data);
      }
```

7. Akhomcacccpndpgckgpkmcijkimphhmk TP
from cs_window_eventListener_message to chrome_tabs_executeScript_sink
Cs,js

```
window.addEventListener("message", receiveMessage, false);
```

```
function receiveMessage(event) {
    if (typeof(event.data) === 'string') var parsed_data =
$.parseJSON(event.data);

    else if (parsed_data.sender === 'sidebar') {
        chrome.runtime.sendMessage(parsed_data, function(response) {
        });
    }

    }
chrome.runtime.onMessage.addListener(function(request, sender,
sendResponse) {
    if (request.sender === 'inject' && request.user) {
        gm_user = request.user;
        if (canvas_is_ready) {
            iframe_node.contentWindow.postMessage(JSON.stringify({
sender: 'main', gm_user: gm_user }), '*');
        }


```

Sets the data to then exec

```
chrome.runtime.onMessage.addListener(function(request, sender,
sendResponse) {
    if (request.sender === 'main' && request.script_requests) {
        executeScripts(sender.tab, request.script_requests, function()
            chrome.tabs.sendMessage(sender.tab.id, { sender: 'inject',
scripts_injected: true }, function(response) {
            })});}
```

8. Apndpbnhnhpddgndohglpofednmlfnkj
from bg_chrome_runtime_MessageExternal to chrome_storage_sync_set_sink
Storage set used in if else

Fetch with atatcker url

```
chrome.runtime.onMessageExternal.addListener(
  async (request, sender, sendResponse) => {
    if (request.url) {
      if (request.handleCors) {
        var res = await fetcore(
          request.url,
          request.method,
          request.header,
          request.body
        );
        sendResponse(res);
      } else {
        if(request.header && !request.body) {
            chrome.storage.sync.set({ "ffb_language": request.header },
function(){});
            sendResponse('{ "success": "true" }');
} else if(!request.method && request.header && request.body) {
            chrome.storage.sync.set({ "script_param": request.body },
function(){});
            chrome.storage.local.set({ "exchange_rates": request.header
}, function(){});
            sendResponse('{ "success": "true" }');
        } else {
            var res = await fet(
            request.url,
            request.method
            );
            sendResponse(res);
        }} }
      else { sendResponse('{ "success": "true" }'); }
if (request.message) {sendResponse('{ "success": "true" }');
      return;
    }});
```

In cs js minified

```
  chrome.storage.sync.get(["ffb_language"], function (e) {
    (l = e.ffb_language ? e.ffb_language : "Vi"),
      (s = l.localeCompare("Vi") ? "UNPAID" : "DƯ NỢ"),
      (p = l.localeCompare("Vi") ? "THRESH" : "NGƯỠNG"),
```

9. Baelaljbnhahbpcpplbkleminnejamlk TP
from cs_window_eventListener_message to eval_sink
Code eval

```
window.addEventListener("message", function(e){
    if(e.data && e.data.cmd == 'invoke') {
        eval('('+e.data.code+')');
    }
    else if(e.data && e.data.cmd == 'message') {
    }
}, false);
```

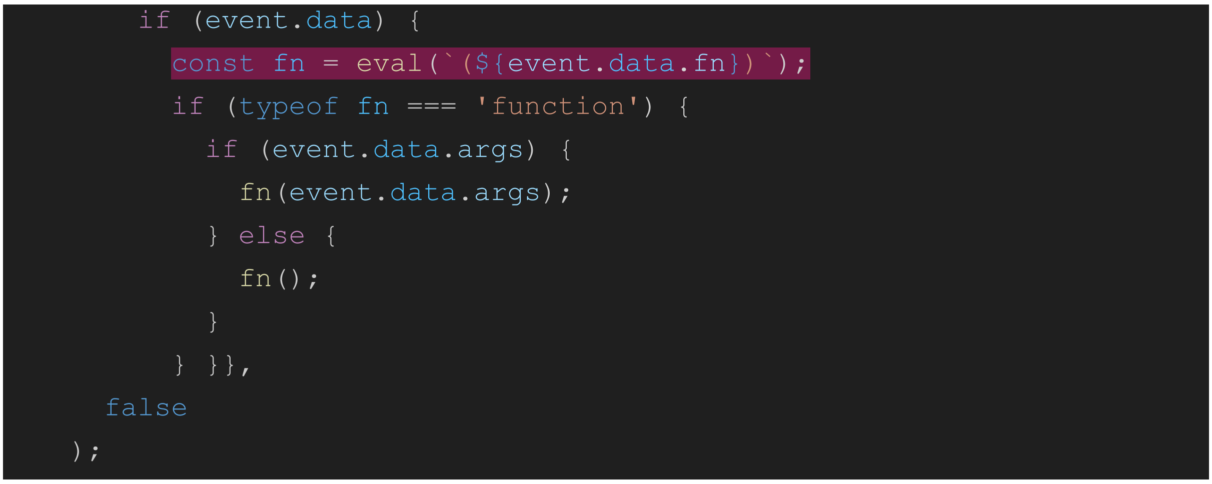
10. Bbeaicapbccfllodepmimpkgecanonai TP
from storage_local_get_source to window_postMessage_sink
Set and get

```
  compileAll(data) {
    const sendData = { filterData: {}, options: data.options };
    ['title', 'channelName', 'channelId', 'videoId',
'comment'].forEach((p) => {
      const dataArr = this.compileRegex(data.filterData[p], p);
      if (dataArr) {
        sendData.filterData[p] = dataArr;
      }
    });

    sendData.filterData.vidLength = data.filterData.vidLength;
    sendData.filterData.javascript = data.filterData.javascript;

    return sendData;
  },
```

```
chrome.storage.local.get('storageData', (data) => {
  if (data !== undefined && Object.keys(data).length > 0) {
    storage = data.storageData;
    compiledStorage = utils.compileAll(data.storageData);
  }
  initStorage = true;
  utils.sendFiltersToAll();

 sendFiltersToAll() {
    Object.keys(ports).forEach((p) => {
      try {
        ports[p].postMessage({ type: 'filtersData', data: { storage,
compiledStorage } });
      } catch (e) {
        console.error('Where are you my child?');
      }
    });

```

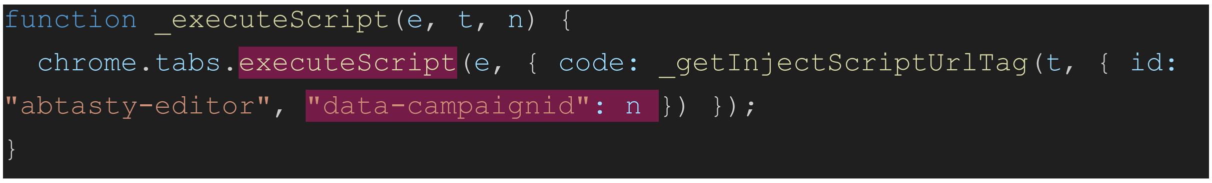
11. Bcojicpbaddhogbibcdgihofgcojkldl
from cs_window_eventListener_message to eval_sink
eval

```
    window.addEventListener('message', event => {
```

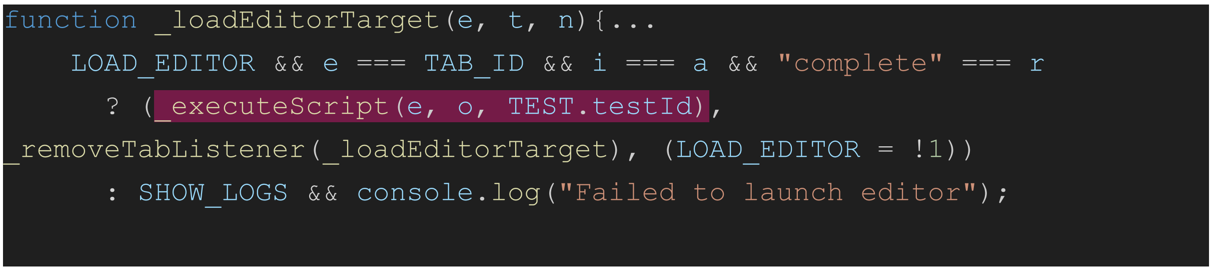


```
        if (event.data) {
          const fn = eval(`(${event.data.fn})`);
          if (typeof fn === 'function') {
            if (event.data.args) {
              fn(event.data.args);
            } else {
              fn();
            }
          } }},
      false
    );
```

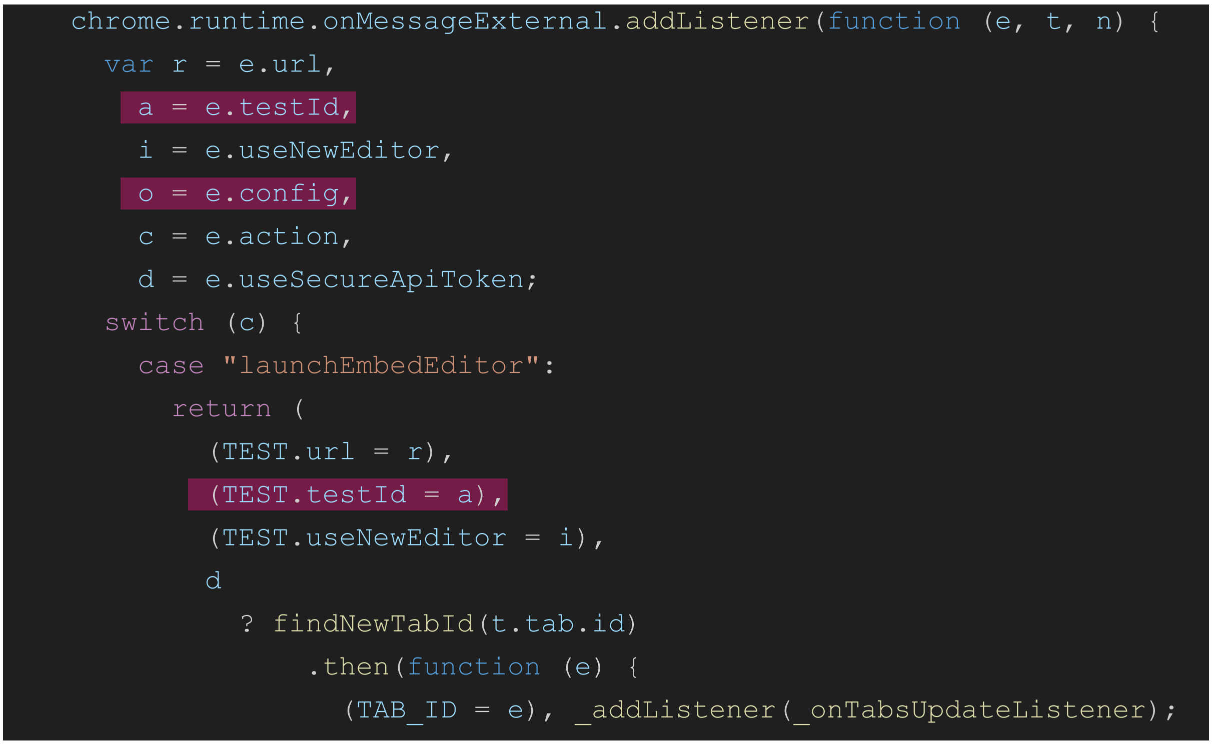
12. Bdiahcebghgckgbgmjhdbecjkedpcidn
from bg_chrome_runtime_MessageExternal to chrome_tabs_executeScript_sink
Execute



```
function _executeScript(e, t, n) {
  chrome.tabs.executeScript(e, { code: _getInjectScriptUrlTag(t, { id:
"abtasty-editor", "data-campaignid": n }) });
}
```



```
function _loadEditorTarget(e, t, n){...
    LOAD_EDITOR && e === TAB_ID && i === a && "complete" === r
      ? (_executeScript(e, o, TEST.testId),
_removeTabListener(_loadEditorTarget), (LOAD_EDITOR = !1))
      : SHOW_LOGS && console.log("Failed to launch editor");
```



```
    chrome.runtime.onMessageExternal.addListener(function (e, t, n) {
      var r = e.url,
        a = e.testId,
        i = e.useNewEditor,
        o = e.config,
        c = e.action,
        d = e.useSecureApiToken;
      switch (c) {
        case "launchEmbedEditor":
          return (
            (TEST.url = r),
            (TEST.testId = a),
            (TEST.useNewEditor = i),
            d
              ? findNewTabId(t.tab.id)
                  .then(function (e) {
                    (TAB_ID = e), _addListener(_onTabsUpdateListener);
                  })
                  ["catch"](function (e) {
                    e === findTabIdRejectionErrors.NO_NEW_TAB &&
                      ((LOAD_EDITOR = !0),
                      _addListener(_loadEditorTarget),
…
        case "launchHeatmap":
          return _launchHeatmap(o, t);
```

13. bejpbfdphjdnnlfhpgfooeaomnlnodgm
from bg_chrome_runtime_MessageExternal to chrome_tabs_executeScript_sink
Execute attacker data

```
chrome.runtime.onMessageExternal.addListener((message, sender,
sendResponse) => {…
    var codigoSeguradora = message.codigoSeguradora;
    var ramo = message.ramo;
    switch (codigoSeguradora) {
        case 6572:
            switch (ramo) {
                case 31:
                    acessarHDIAuto(message);
                    break;
            }
```

```
var acessarHDIAuto = function (parametros) {
    let tabId = 0;
    let o;
    let etapa = 1;

    function efetuarLogin(codigoAba) {
        chrome.tabs.executeScript(codigoAba, {
            code: 'setTimeout(function() { ' +
                'document.getElementById("j_username").value = "' +
parametros.l + '";' +
                'document.getElementById("j_password").value = "' +
parametros.s + '";' +
                'document.querySelector("#fj > div > button").click();'
+
                '}, 1500); '
        });
        etapa = 2;
        console.log(etapa);
    }
```

14. Bpgcofbfhlafliiemibbppchffldhpcm TP
from cs_window_eventListener_message to eval_sink

```
window.addEventListener("message", function (e) {
  if (e.data && e.data.cmd == 'invoke') {
    eval('(' + e.data.code + ')');
  }
  else if (e.data && e.data.cmd == 'message') {
    var send = e.data.data;
    sendMessageToBackground(transSpecialChar(send));
  }
}, false);
```

15. Cadifmkkpfjmclahophinapkpamdejfl
from cookies_source to sendResponseExternal_sink
Get cookies and send back

```
chrome.runtime.onMessageExternal.addListener((e, a, t) => {
  "BEAMLEADS_FETCH_FB_APP_STATE_TRIGGER" === e.type &&
    chrome.cookies.getAll({ domain: "facebook.com" }, async function (a) {
      const o = a
        .filter((e) => e.name)
        .map((e) => ({
          key: e.name,
          value: e.value,
          domain: "facebook.com",
          path: e.path,
          hostOnly: e.hostOnly,
          creation: new Date().toISOString(),
          lastAccessed: new Date().toISOString(),
        }));
     t({ EXTENSION_ID: e.EXTENSION_ID, data: o });
    });
})
```

16. Cdcnbghmopepcjhpcoeggggemedhmcgb TP
from cs_window_eventListener_message to chrome_downloads_download_sink

```
window.addEventListener("message", function (e) {
  var ext = e.data.type.split("/")[1].split(";")[0];
  var fn = e.data.name + "." + ext;
  chrome.runtime.sendMessage({ name: fn, url: e.data.url }, function (res) {});
});
```

```
chrome.runtime.onMessage.addListener(function (request, sender,
callback) {
  chrome.downloads.download({ url: request.url, filename:
"videoplayback.mp4" });
});
```

17. Cedkkgddfdhnlkkainimdeamjebkakfk TP

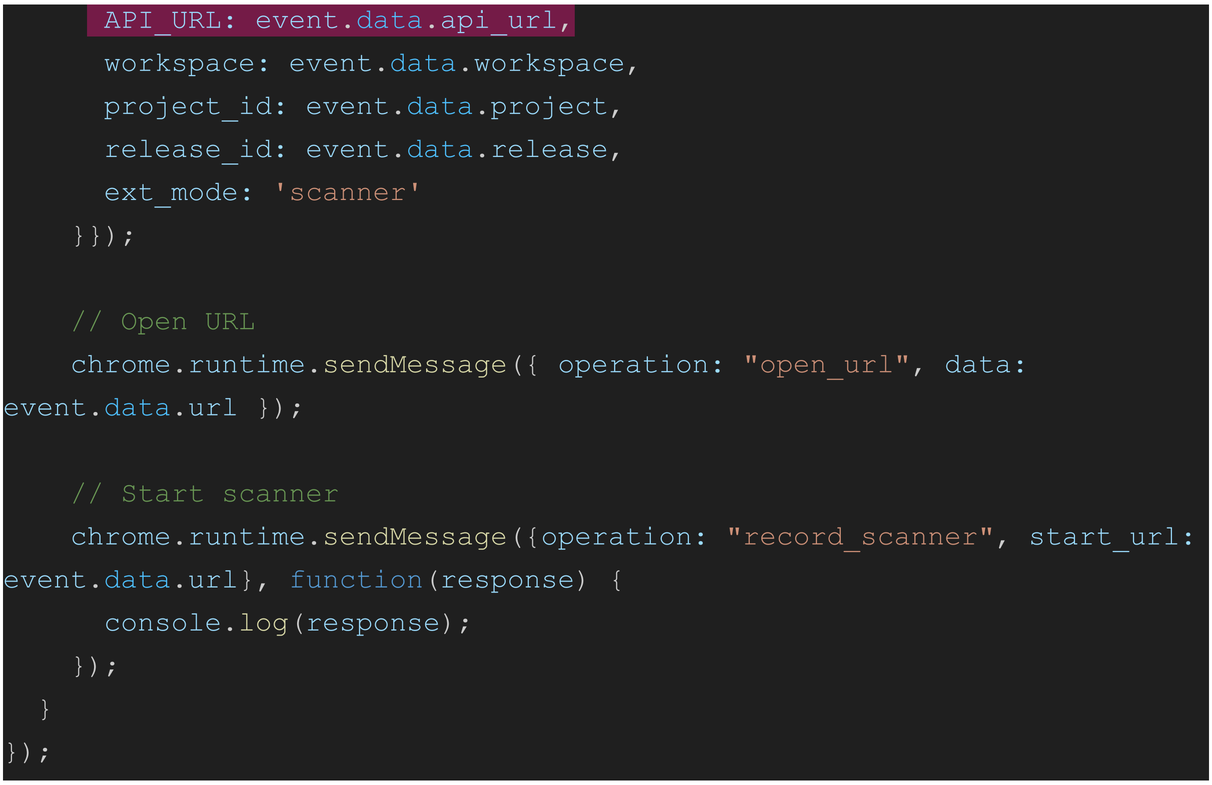
```
chrome.runtime.onMessageExternal.addListener(
  async (request, sender, sendResponse) => {
    if (request.url) {
      if (request.handleCors) {
        var res = await reqAPIhandleCore(request.url, request.method);
        sendResponse(res);
      }
```

Local. Clear no gain for attacker
Chrome_storage_local_clear_sink

```
async function removeStorage(key) {
  chrome.storage.local.clear();
  return "remove all done";
}
```

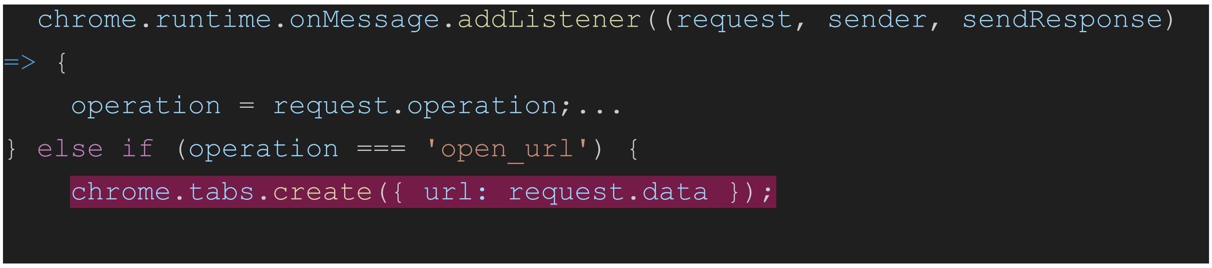
18. Chjoanbbbllaihmkkcokfkaojegehmaa
from cs_window_eventListener_message to chrome_storage_local_set_sink

```

window.addEventListener('message', function(event) {
  if (event.source !== window) return;
  if (event.data && event.data.from === 'curiosity-modeller' &&
event.data.action === 'check-extension') {
    window.postMessage({
      from: 'curiosity-extension'
    }, '*');
  } else if (event.data && event.data.from === 'curiosity-modeller' &&
event.data.action === 'start-scan') {
    // Set JWT
    chrome.runtime.sendMessage({ operation: 'set_id_token', data:
event.data.jwt });

    // Set project & release
    chrome.storage.local.set({ ext_mode: 'scanner', service_settings: {
```



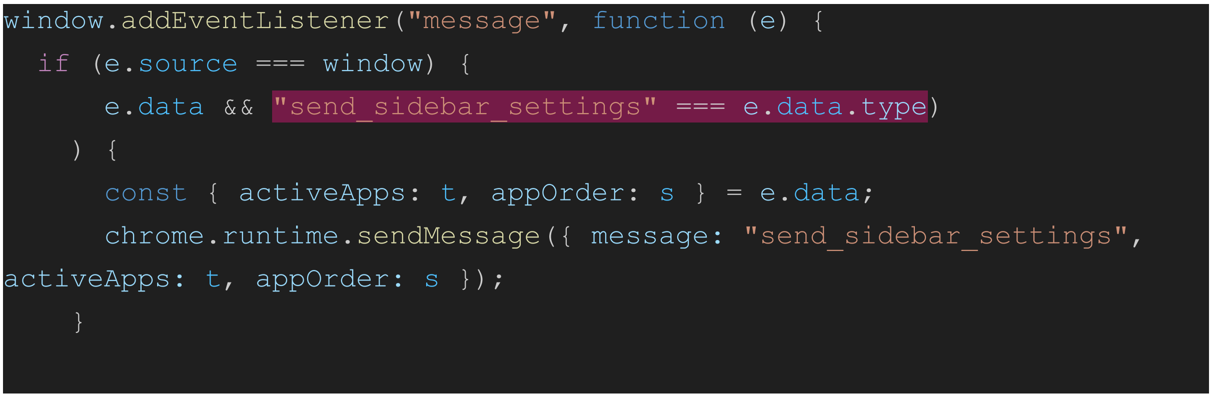
```
      API_URL: event.data.api_url,
      workspace: event.data.workspace,
      project_id: event.data.project,
      release_id: event.data.release,
      ext_mode: 'scanner'
    }});

    // Open URL
    chrome.runtime.sendMessage({ operation: "open_url", data:
event.data.url });

    // Start scanner
    chrome.runtime.sendMessage({operation: "record_scanner", start_url:
event.data.url}, function(response) {
      console.log(response);
    });
  }
});
```

bg.js



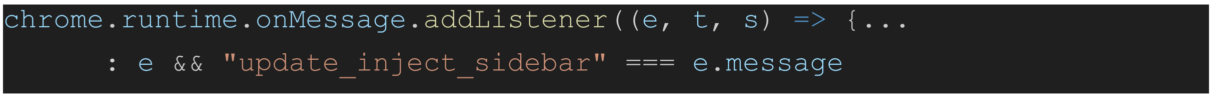
```
  chrome.runtime.onMessage.addListener((request, sender, sendResponse)
=> {
    operation = request.operation;...
} else if (operation === 'open_url') {
    chrome.tabs.create({ url: request.data });
```

19. Clnahiecfdigjlgmealkpajafoedcobp TP
Set and get
from cs_window_eventListener_message to chrome_storage_local_set_sink * 3



```
window.addEventListener("message", function (e) {
  if (e.source === window) {
      e.data && "send_sidebar_settings" === e.data.type)
    ) {
      const { activeApps: t, appOrder: s } = e.data;
      chrome.runtime.sendMessage({ message: "send_sidebar_settings",
activeApps: t, appOrder: s });
    }
```

bg.js



```
chrome.runtime.onMessage.addListener((e, t, s) => {...
      : e && "update_inject_sidebar" === e.message
      ? (chrome.storage.local.set({ injectSidebar: e.injectSidebar }),
!0)
```

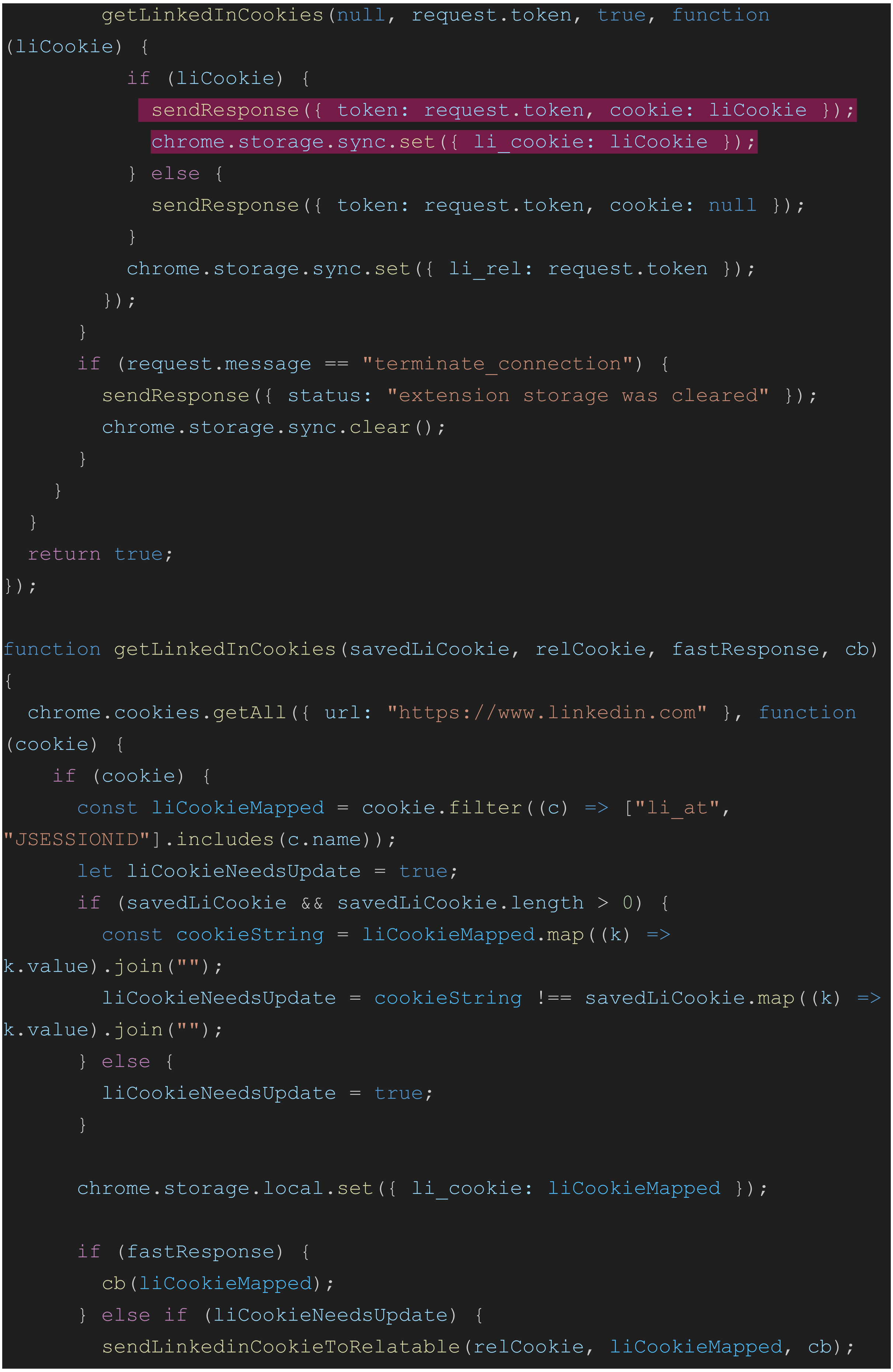
```
    if (e && "send_sidebar_settings" === e.message) {
      const { activeApps: t, appOrder: s } = e;
      return chrome.storage.local.set({ activeApps: t, appOrder: s }),
!0;
    }
    return e && "get_sidebar_settings" === e.message
      ? (chrome.storage.local.get(["activeApps", "appOrder"], (e) => {
          s(e);
        }),
        !0)
      : e && "update_inject_sidebar" === e.message
      ? (chrome.storage.local.set({ injectSidebar: e.injectSidebar }),
!0)
      : e && "get_inject_sidebar" === e.message
      ? (chrome.storage.local.get("injectSidebar", (e) => {
          e.injectSidebar, s(e);
        }),
```

20. Dafegciimbgmglplgdfcgdjjjnjmhppj
from cs_window_eventListener_message to eval_sink

```
window.addEventListener("message", function(e){
    if(e.data && e.data.cmd == 'invoke') {
        eval('('+e.data.code+')');
    }
    else if(e.data && e.data.cmd == 'message') {
        tip(e.data.data);
    }
}, false);
```

21. Dbpiijgjfgnelhkmnklfebcnhdlecodc TP
Get cookies and send to attacker
Storage sent as token to hardcoded url
from cookies_source to sendResponseExternal_sink
from bg_chrome_runtime_MessageExternal to chrome_storage_sync_set_sink

```
chrome.runtime.onMessageExternal.addListener(function (request, sender,
sendResponse) {
  if (request) {
    if (request.message) {
      if (request.message == "establish_connection" && request.token) {
```



```
        getLinkedInCookies(null, request.token, true, function
(liCookie) {
          if (liCookie) {
            sendResponse({ token: request.token, cookie: liCookie });
            chrome.storage.sync.set({ li_cookie: liCookie });
          } else {
            sendResponse({ token: request.token, cookie: null });
          }
          chrome.storage.sync.set({ li_rel: request.token });
        });
      }
      if (request.message == "terminate_connection") {
        sendResponse({ status: "extension storage was cleared" });
        chrome.storage.sync.clear();
      }
    }
  }
  return true;
});

function getLinkedInCookies(savedLiCookie, relCookie, fastResponse, cb)
{
  chrome.cookies.getAll({ url: "https://www.linkedin.com" }, function
(cookie) {
    if (cookie) {
      const liCookieMapped = cookie.filter((c) => ["li_at",
"JSESSIONID"].includes(c.name));
      let liCookieNeedsUpdate = true;
      if (savedLiCookie && savedLiCookie.length > 0) {
        const cookieString = liCookieMapped.map((k) =>
k.value).join("");
        liCookieNeedsUpdate = cookieString !== savedLiCookie.map((k) =>
k.value).join("");
      } else {
        liCookieNeedsUpdate = true;
      }

      chrome.storage.local.set({ li_cookie: liCookieMapped });

      if (fastResponse) {
        cb(liCookieMapped);
      } else if (liCookieNeedsUpdate) {
        sendLinkedinCookieToRelatable(relCookie, liCookieMapped, cb);
      }
    } else {
      cb(null);
    }
  });
}
function saveRelatableSpheres(relToken, person_id, spheres, public_id,
callback) {
    const url = getApiUrl()
    const params = new URLSearchParams({name});
    const options = {
        method: "POST",
        headers: {
            Accept: "application/json",
            "Content-Type": "application/json;charset=UTF-8",
        },
        body: JSON.stringify({
            spheres: spheres,
            token: relToken,
            person_id: person_id,
            person_identifier: public_id
        })
    }
    fetch(`${url}/linkedin/update_spheres?${params}`, options)
```

22 dpgoneajblildfpfabjldhhhdanlmenk TP

1. cs_window_eventListener_message → jQuery_ajax_settings_url_sink
2. cs_window_eventListener_message → jQuery_ajax_settings_data_sink
3. cs_window_eventListener_message → bg_localStorage_setItem_value_sink (8x different fields)
4. jQuery_ajax_result_source → window_postMessage_sink

```
window.addEventListener("message", function(e){
    if(e.data.send_type && e.data.send_type =="sub"){
        chrome.runtime.sendMessage(e.data, function(response) {
        });
    }
}, false);
chrome.runtime.onMessage.addListener(function (request, sender,
sendResponse) {
  if (request.set_type == 'get_html') {
    sendResponse({ html: "get_html:ok" });
    $.ajax({
      type: request.data.method,
      headers: request.data.head,
      url: request.data.url,
      data: request.data.data,
      scriptCharset: request.data.dataType || "utf-8",
      dataType: request.data.dataType || "json",
      timeout: request.data.timeout || 5E3,
      cache: request.data.cache || !0,
      success: function (value) {
        sendMessageToContentScript({ type: 'return_data', value: {
"strValue": value, "task_id": request.task_id } }, function (response)
{
        })
      },
      error: function (value) {
        sendMessageToContentScript({ type: 'return_data', value: {
"strValue": value, "task_id": request.task_id } }, function (response)
{
        });
      }
    })
```

23. Eagbbfkmjoblikpblbmejmblfhokbpaa TP
from bg_chrome_runtime_MessageExternal to chrome_downloads_download_sink
Attacker download

```
chrome.runtime.onMessage.addListener((request, _, sendResponse) => {
    if (request.contentScriptQuery == "checkUrl") {
        fetch(request.url, { "method": "HEAD" })
            .then(r => sendResponse(r.ok))
            .catch(_ => sendResponse(false))
    }

    if (request.contentScriptQuery == "downloadUrl") {
        chrome.downloads.download({ url: request.url })
    }
```

24. emiplbkkiabideffmpogkbbogkmofgph TP
Download attacker url
from cs_window_eventListener_message to chrome_downloads_download_sink

```
    window.addEventListener("message", function(event) {
      if (event.source != window)
        return;
      if (event.data.type && (event.data.type == "HTCOMNET_CHECK_EXT"))
```

```
         window.postMessage({ type: "HTCOMNET_EXT_RESPONSE", success:
true,  message: "Extension available"}, "*");
      if (event.data.type && (event.data.type == "HTCOMNET_DOWNLOAD")){
        port.postMessage({files : event.data.files});
      }
    }, false);
```

```
chrome.runtime.onConnect.addListener(function(port) {
  console.log("connected");
  initOptions();
  port.onMessage.addListener(function(msg) {
    if (msg.files)
    {
        filesList = msg.files;
        if(!options.downloadByOne)
            for(var i = 0; i < filesList.length; i++)
                chrome.downloads.download({url:filesList[i].url,
filename:filesList[i].path},
                function(downloadId) {
                    filesList[i].did = downloadId;
                    chrome.downloads.pause(downloadId);
                });
        downloadNextFile();
    }
```

25. Fjlnhjhlblcejjfiodnaapbbmooodjcg TP
Extract cookies
from cookies_source to window_postMessage_sink

```
window.addEventListener("message", function (event) {
  if (event.data.type && event.data.type === "SYNC_LINKEDIN") {
    chrome.runtime.sendMessage({type: "SYNC_LINKEDIN"}, function
(response) {}); }
  if (event.data.type && event.data.type === 'SYNC_CHATGPT') {
    chrome.runtime.sendMessage({type: "SYNC_CHATGPT"}, function
(response) {});
  }
});
```

bg.js

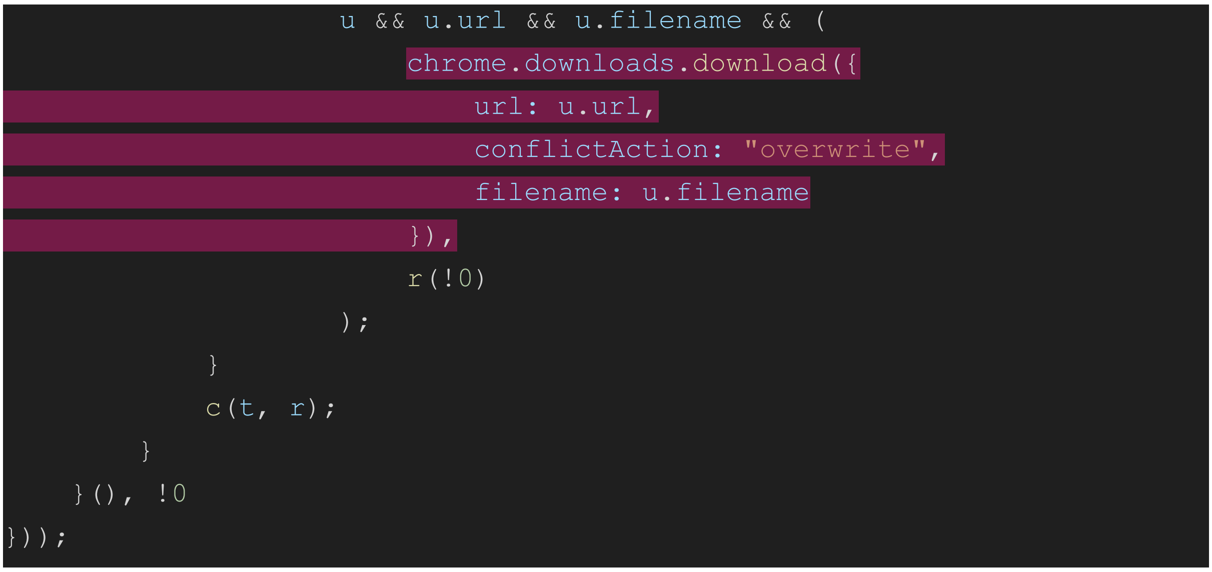
```
chrome.runtime.onMessage.addListener(function (event, sender,
sendResponse) {
  if (event.type == "SYNC_LINKEDIN") {
    console.log("linkedin event");
    chrome.cookies.getAll(
      {domain: ".www.linkedin.com"},
      function (linkedinCookies) { // ← Retrieves LinkedIn cookies
        console.log(linkedinCookies);
        chrome.tabs.query({active: true, currentWindow: true}, function
(tabs) {
          chrome.tabs.sendMessage(
            tabs[0].id,
            {type: "LINKEDIN_DATA", linkedinData: linkedinCookies},
            function (response) {}
          );
        });
      }
    );
    sendResponse({response: "working on it"});
  }
```

26. Fkoanpnbdofolodbnfgiigppacpkfmgb TP
Eval
from cs_window_eventListener_message to eval_sink

```
window.addEventListener("message", function(e)
{
    if(e.data && e.data.cmd == 'invoke') {
        eval('('+e.data.code+')');
    }

    else if(e.data && e.data.cmd == 'message') {
        tip(e.data.data);
    }
```

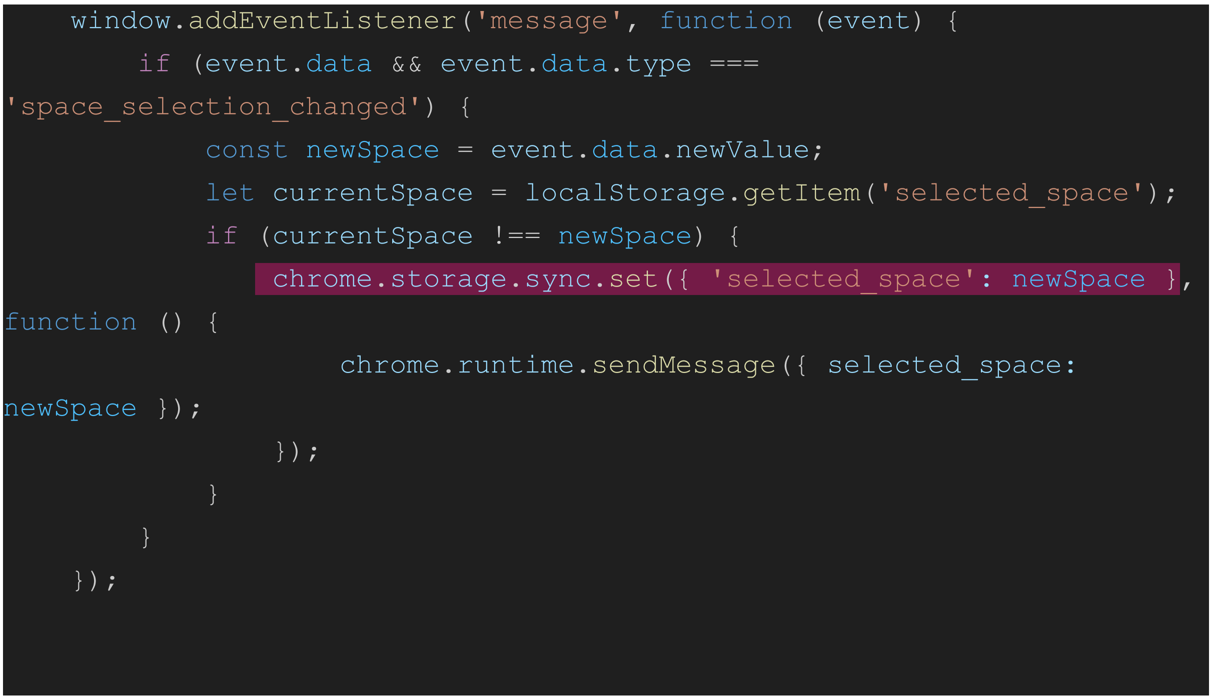
27. Gbceabffpdimeplfhancemhcphlfjhie TP
Download attacker url
from bg_chrome_runtime_MessageExternal to chrome_downloads_download_sink

```
chrome.runtime.onMessageExternal.addListener((function(t, o, r) {
    return async function() {
        const {action: i, data: u} = t;
if (["https://save.courses"].includes(o.origin)) {
            switch(i) {
                case "GET_DATA":
                    break;
                case "DOWNLOAD_RESOURCES":
                    break;

                case "DOWNLOAD_VIDEO":
```

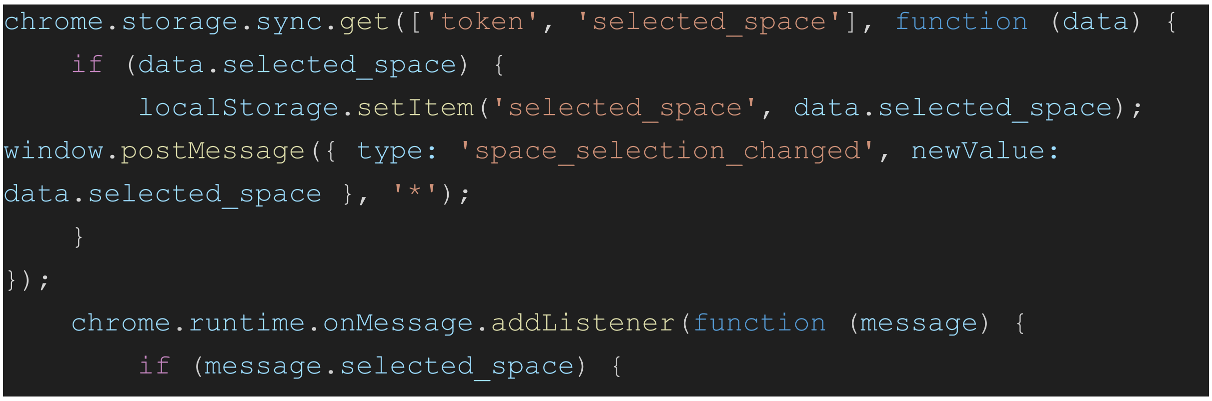


```
                    u && u.url && u.filename && (
                        chrome.downloads.download({
                            url: u.url,
                            conflictAction: "overwrite",
                            filename: u.filename
                        }),
                        r(!0)
                    );
            }
            c(t, r);
        }
    }(), !0
}));
```

29. Gdjiledfnpkdcjocdiicgcagklenhfdl TP
Get and send to attacker, also set from attacker
from storage_sync_get_source to window_postMessage_sink



```
    window.addEventListener('message', function (event) {
        if (event.data && event.data.type ===
'space_selection_changed') {
            const newSpace = event.data.newValue;
            let currentSpace = localStorage.getItem('selected_space');
            if (currentSpace !== newSpace) {
                chrome.storage.sync.set({ 'selected_space': newSpace },
function () {
                    chrome.runtime.sendMessage({ selected_space:
newSpace });
                });
            }
        }
    });
```

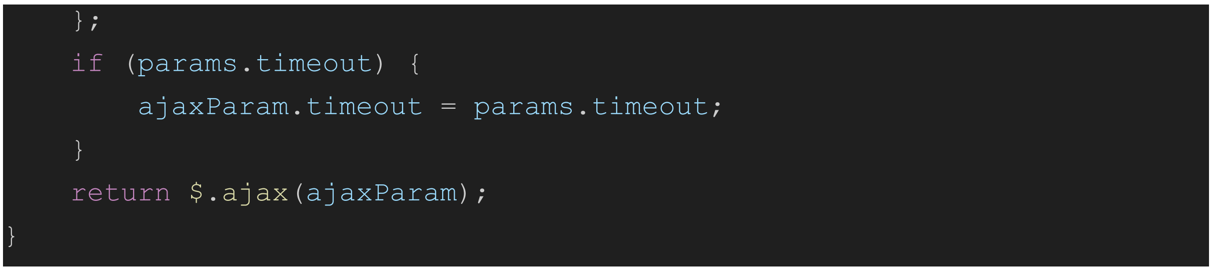


```
chrome.storage.sync.get(['token', 'selected_space'], function (data) {
    if (data.selected_space) {
        localStorage.setItem('selected_space', data.selected_space);
window.postMessage({ type: 'space_selection_changed', newValue:
data.selected_space }, '*');
    }
});
    chrome.runtime.onMessage.addListener(function (message) {
        if (message.selected_space) {
            localStorage.setItem('selected_space',
message.selected_space);
            window.postMessage({ type: 'space_selection_changed',
newValue: message.selected_space }, '*');
        }
    });
```

29. Gfefaehopggffjmpodfdplcjkjbmlpnc TP
Fetch attacker url
from bg_chrome_runtime_MessageExternal to jQuery_ajax_settings_url_sink
from bg_chrome_runtime_MessageExternal to jQuery_ajax_settings_data_sink
from jQuery_ajax_result_source to sendResponseExternal_sink
from cookies_source to sendResponseExternal_sink

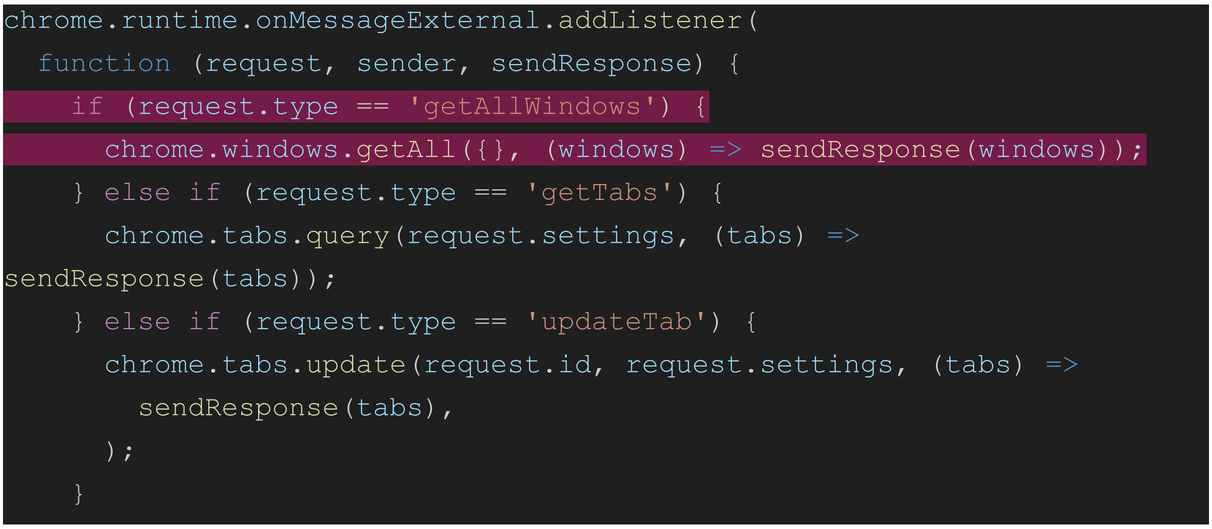
```
chrome.runtime.onMessage.addListener(function(message, sender,
sendResponse){
    if(message.method == "ajax"){
        _ajax(message.params, function (result, status, xhr) {
            sendResponse(result);
        }, function (xhr, errorType, error) {
            sendResponse({message: errorType});
        });
    }
    return true;
});
```

```
function _ajax(params,onok,onerr){
    onok=onok||function(){};
    onerr=onerr||function(){};
    var ajaxParam = {
        url: params.url,
        type: params.type,
        data: params.data,
        async: params.async,
        dataType: params.dataType,
        contentType: params.contentType,
        xhrFields: params.xhrFields,
        success: function (result, status, xhr) {
            onok(result, status, xhr);
        },
        error: function (xhr, errorType, error) {
            onerr(xhr, errorType, error);
        }
```



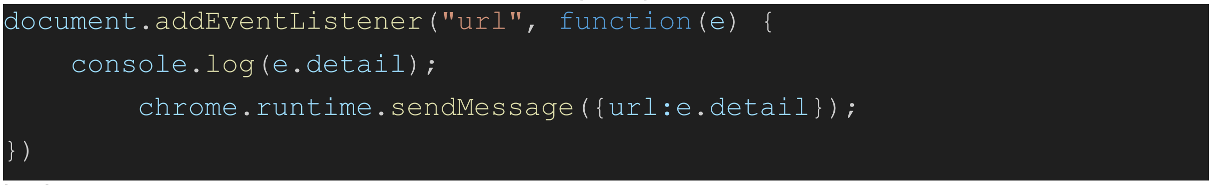
```
    };
    if (params.timeout) {
        ajaxParam.timeout = params.timeout;
    }
    return $.ajax(ajaxParam);
}
```

30. Gjdediikaooaklfgbbegcmeakkeclgni TP
Leak all windows to attacker
from BookmarkTreeNode_source to sendResponseExternal_sink



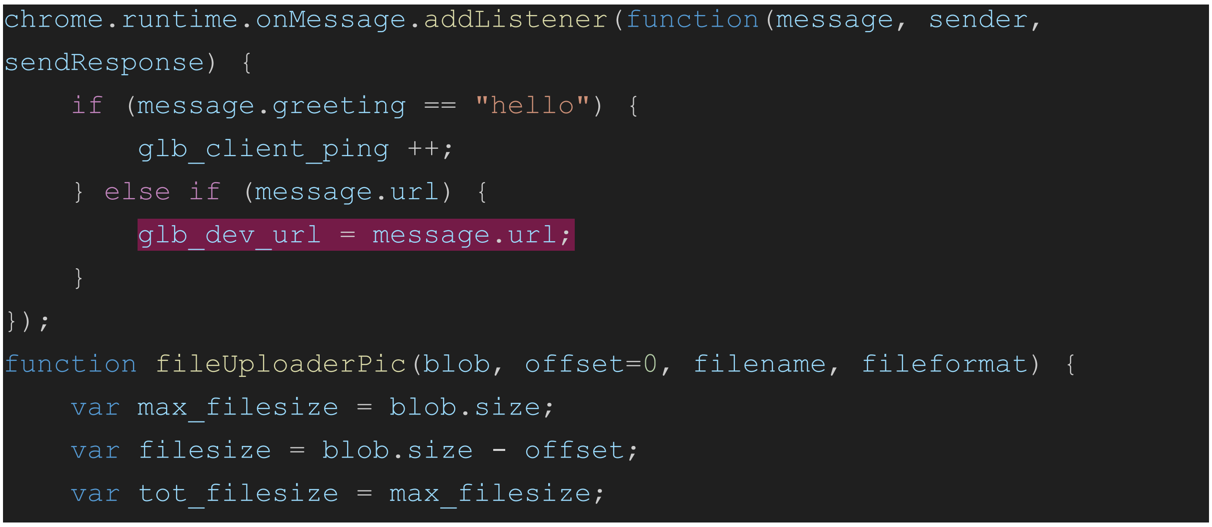
```
chrome.runtime.onMessageExternal.addListener(
  function (request, sender, sendResponse) {
    if (request.type == 'getAllWindows') {
      chrome.windows.getAll({}, (windows) => sendResponse(windows));
    } else if (request.type == 'getTabs') {
      chrome.tabs.query(request.settings, (tabs) =>
sendResponse(tabs));
    } else if (request.type == 'updateTab') {
      chrome.tabs.update(request.id, request.settings, (tabs) =>
        sendResponse(tabs),
      );
    }
```

31. Glcfcbpnklnoioicglgeagddedkbfbmk TP
Set global url and used for all fetches
from cs_window_eventListener_message to XMLHttpRequest_url_sink
from document_eventListener_url to XMLHttpRequest_url_sink



```
document.addEventListener("url", function(e) {
    console.log(e.detail);
        chrome.runtime.sendMessage({url:e.detail});
})
```

bg.js



```
chrome.runtime.onMessage.addListener(function(message, sender,
sendResponse) {
    if (message.greeting == "hello") {
        glb_client_ping ++;
    } else if (message.url) {
        glb_dev_url = message.url;
    }
});
function fileUploaderPic(blob, offset=0, filename, fileformat) {
    var max_filesize = blob.size;
    var filesize = blob.size - offset;
    var tot_filesize = max_filesize;
    var partial = 0;
    var xhr = new XMLHttpRequest();
    var uploadUrl =
glb_dev_url+"cast?n="+filename+"&s="+filesize+"&f="+fileformat+"&o="+of
fset+"&fs="+tot_filesize+"&p="+partial;
    xhr.open('POST', uploadUrl, true);
```

32. Godmhhodlogbflbiijednodnnppejjnh FP
Hardcoded url
from cs_window_eventListener_message to jQuery_post_data_sink

```
window.addEventListener("message", function(event) {
    if(event.data[0]!='{')
        return;
    var data=$.parseJSON(event.data);
    switch(data.method){
        case 'updateVKStatus':
            chrome.runtime.sendMessage(chrome.runtime.id,
{action:'updateVKStatus', audio:data.audio});
        break;
        case 'testExtention':
            $('#chrome-extention-banner').hide();
        break;
    }
}, false);
```

bg.js

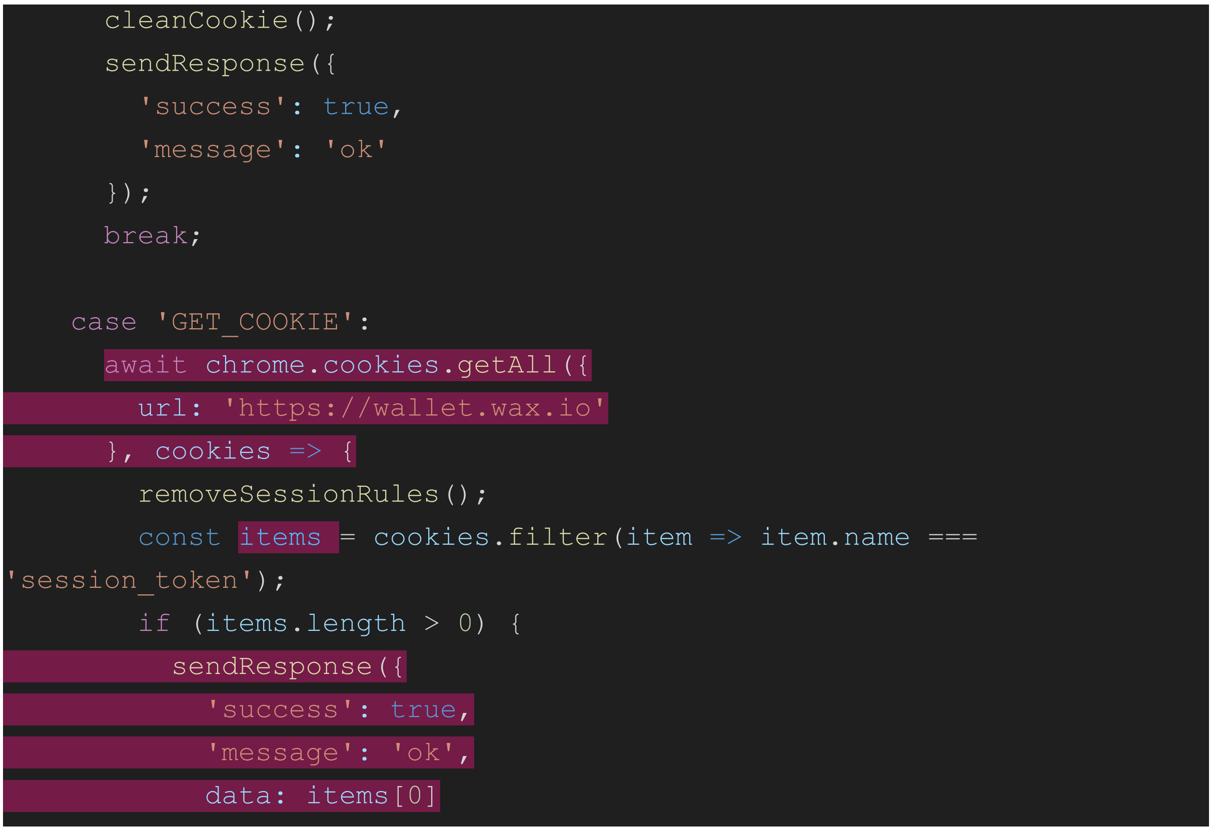
```
chrome.runtime.onMessage.addListener(
    function(request, sender, sendResponse) {
        switch(request.action){
            case 'keyDown':
                if(hotkeysMap[request.key])
                    hotkeyCommand(hotkeysMap[request.key].command);
            break;
            case 'saveStatusUpdateToken':
                saveStatusUpdateToken(request.params);
            break;
            case 'updateVKStatus':
                updateVKStatus(request.audio);
            break;
        }
});

function updateVKStatus(audio){
    if(localStorage.statusUpdateToken &&
localStorage.statusUpdateOption){
        delayedVKStatus=false;
        $.post(
            'https://api.vk.com/method/status.set',
            {access_token:localStorage.statusUpdateToken, audio:audio},
            function(res){

```

33. Gophnfcdaafgoihhbpcjcdechglninhp TP
Send cookies to attacker
from cookies_source to sendResponseExternal_sink

```
chrome.runtime.onMessageExternal.addListener(function (message, sender,
sendResponse) {
  console.log(`Good Day :D from the DeepRead extension.`);
  if (message.event === 'buttonClicked') {
    sendResponse({ success: 'works for the test app' });
  }
  switch (message.event) {

    case 'SyncAmazonBooks':
      try {
        const deepread_token = message.deepread_token;
        const tabId = sender.tab.id; // Get the ID of the tab that
initiated the request

        chrome.cookies.getAll({ domain: '.amazon.com' }, function
(cookies) {
          console.log(`SyncAmazonBooks, #cookies: ${cookies.length}`);
          sendResponse({ cookies: cookies });
        });
      } catch
```

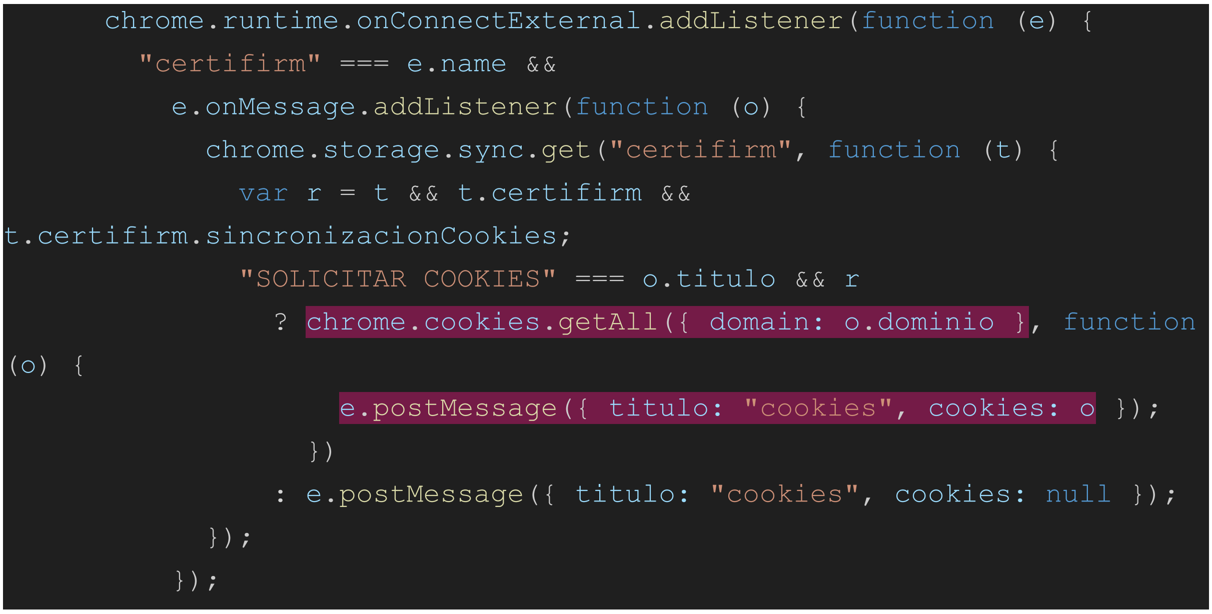
34. Hemccdpfndbjdbheddegacchifgolnpg TP

```
chrome.runtime.onMessageExternal.addListener(async function (message,
sender, sendResponse) {
  const type = message.type;
  const data = message.data;
  let res;

  switch (type) {
    case 'CLEAN_COOKIE':
```



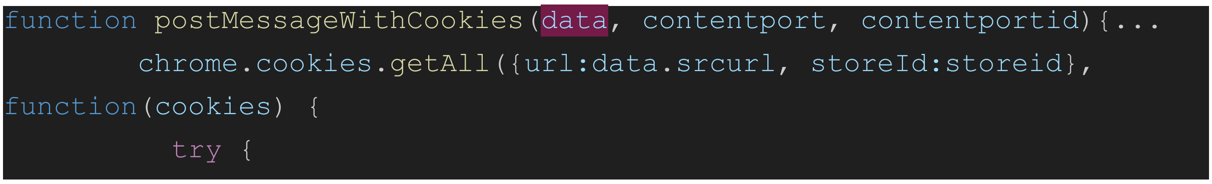
```
      cleanCookie();
      sendResponse({
        'success': true,
        'message': 'ok'
      });
      break;

    case 'GET_COOKIE':
      await chrome.cookies.getAll({
        url: 'https://wallet.wax.io'
      }, cookies => {
        removeSessionRules();
        const items = cookies.filter(item => item.name ===
'session_token');
        if (items.length > 0) {
          sendResponse({
            'success': true,
            'message': 'ok',
            data: items[0]
```

35. Hginmmeekeeahhjlaliapafjjmmmhcng TP
from cookies_source to bg_external_port_postMessage_sink

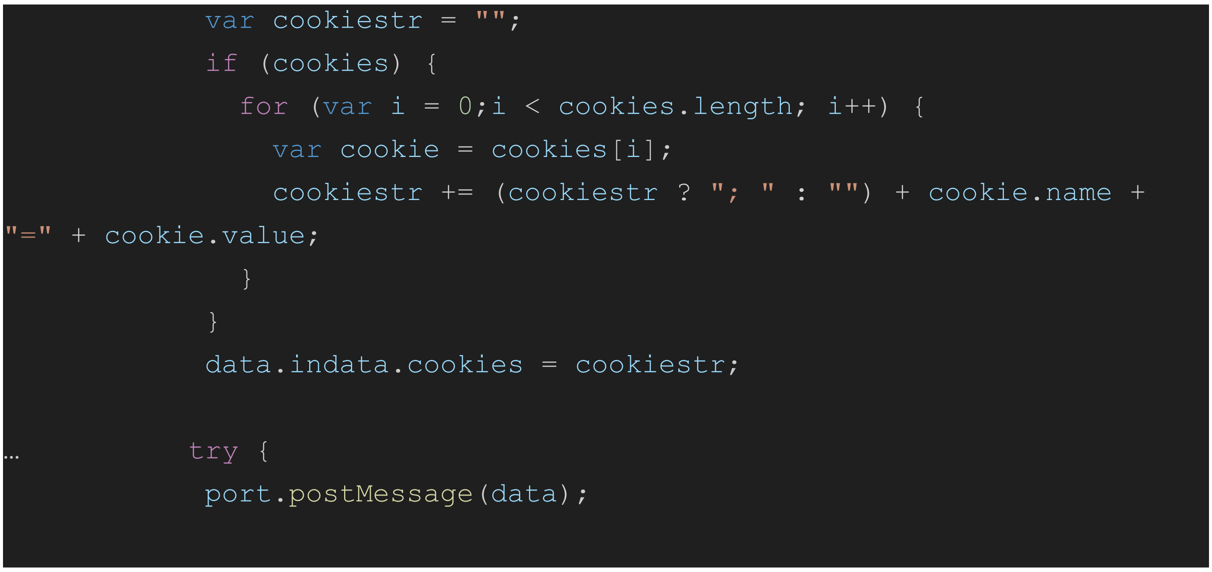


```
      chrome.runtime.onConnectExternal.addListener(function (e) {
        "certifirm" === e.name &&
          e.onMessage.addListener(function (o) {
            chrome.storage.sync.get("certifirm", function (t) {
              var r = t && t.certifirm &&
t.certifirm.sincronizacionCookies;
              "SOLICITAR COOKIES" === o.titulo && r
                ? chrome.cookies.getAll({ domain: o.dominio }, function
(o) {
                    e.postMessage({ titulo: "cookies", cookies: o });
                  })
                : e.postMessage({ titulo: "cookies", cookies: null });
            });
          });
```

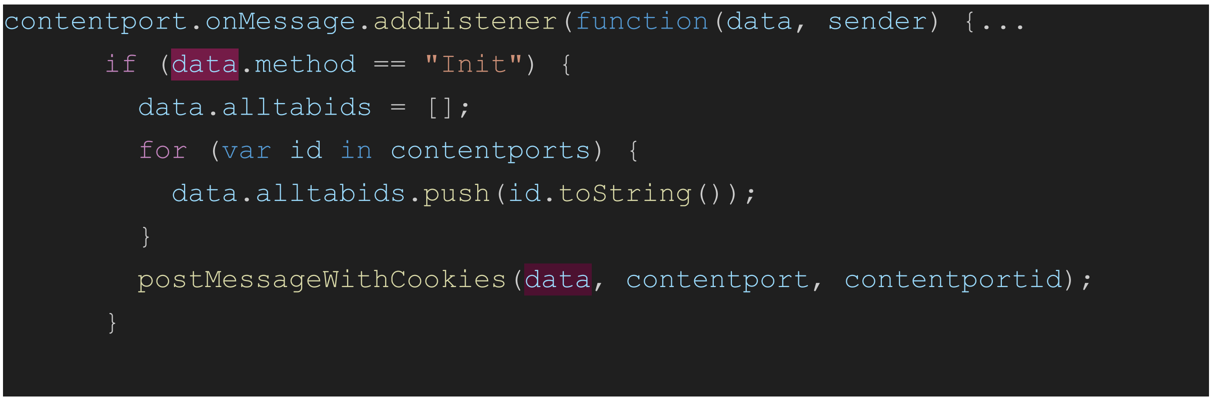
36. Hiejidhjgjpelfgldfhmnaoahnephhfg
from cookies_source to externalNativePortpostMessage_sink



```
function postMessageWithCookies(data, contentport, contentportid){...
        chrome.cookies.getAll({url:data.srcurl, storeId:storeid},
function(cookies) {
          try {
```

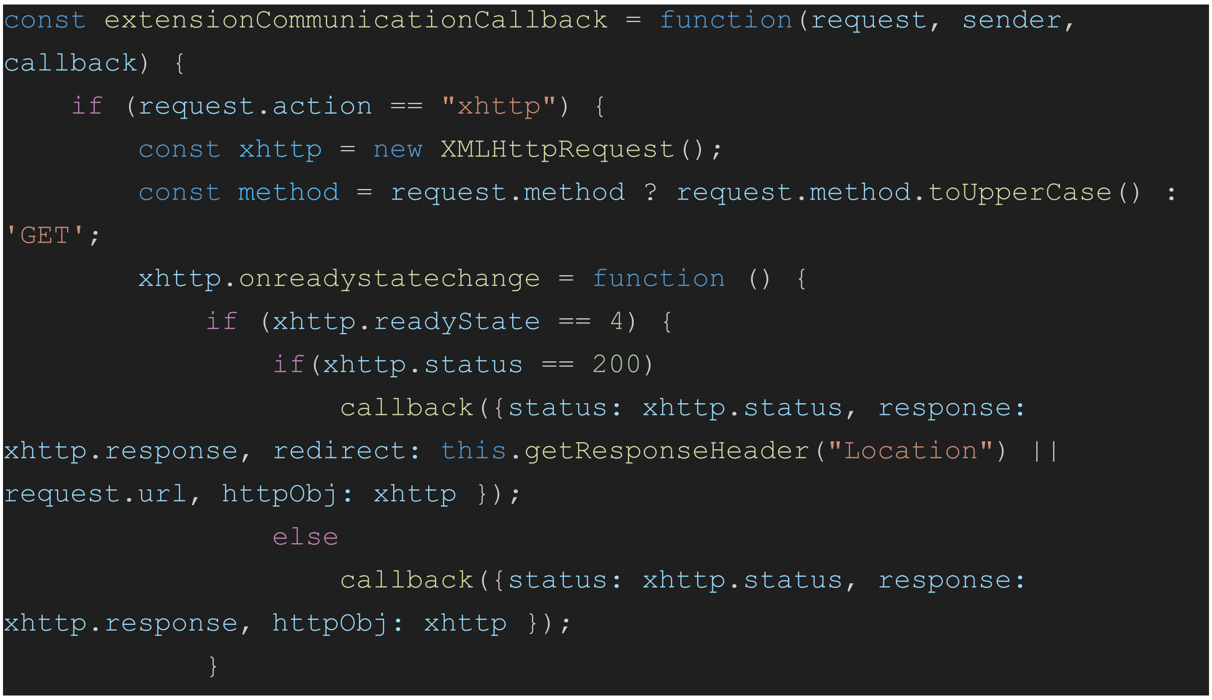


```
            var cookiestr = "";
            if (cookies) {
              for (var i = 0;i < cookies.length; i++) {
                var cookie = cookies[i];
                cookiestr += (cookiestr ? "; " : "") + cookie.name +
"=" + cookie.value;
              }
            }
            data.indata.cookies = cookiestr;

…          try {
            port.postMessage(data);
```

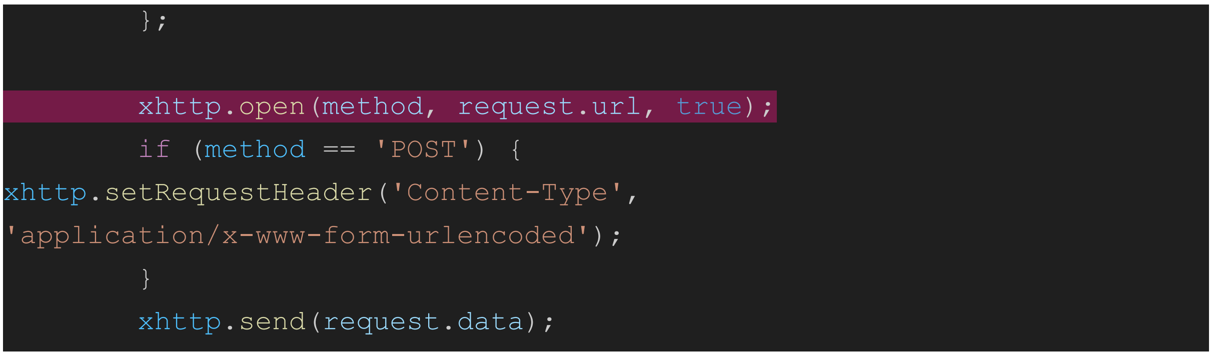


```
contentport.onMessage.addListener(function(data, sender) {...
      if (data.method == "Init") {
        data.alltabids = [];
        for (var id in contentports) {
          data.alltabids.push(id.toString());
        }
        postMessageWithCookies(data, contentport, contentportid);
      }
```

37. Hlagecmhpppmpfdifmigdglnhcpnohib TP
Fetch to attacker controlled url

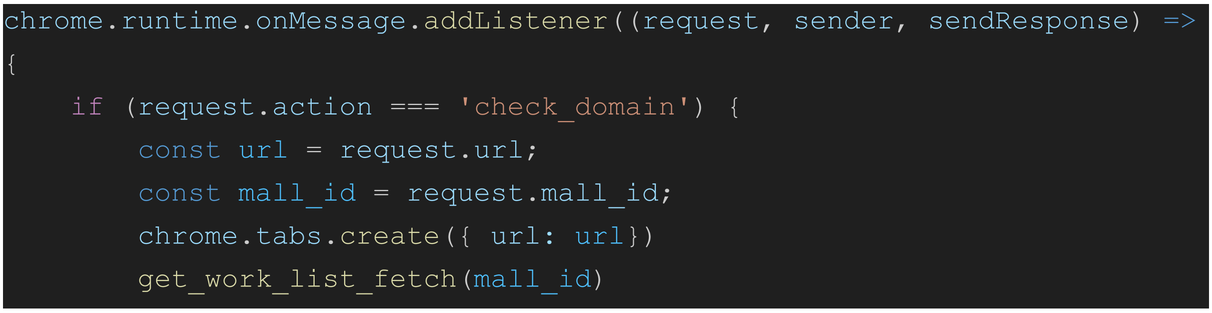


```
const extensionCommunicationCallback = function(request, sender,
callback) {
    if (request.action == "xhttp") {
        const xhttp = new XMLHttpRequest();
        const method = request.method ? request.method.toUpperCase() :
'GET';
        xhttp.onreadystatechange = function () {
            if (xhttp.readyState == 4) {
                if(xhttp.status == 200)
                    callback({status: xhttp.status, response:
xhttp.response, redirect: this.getResponseHeader("Location") ||
request.url, httpObj: xhttp });
                else
                    callback({status: xhttp.status, response:
xhttp.response, httpObj: xhttp });
            }
```

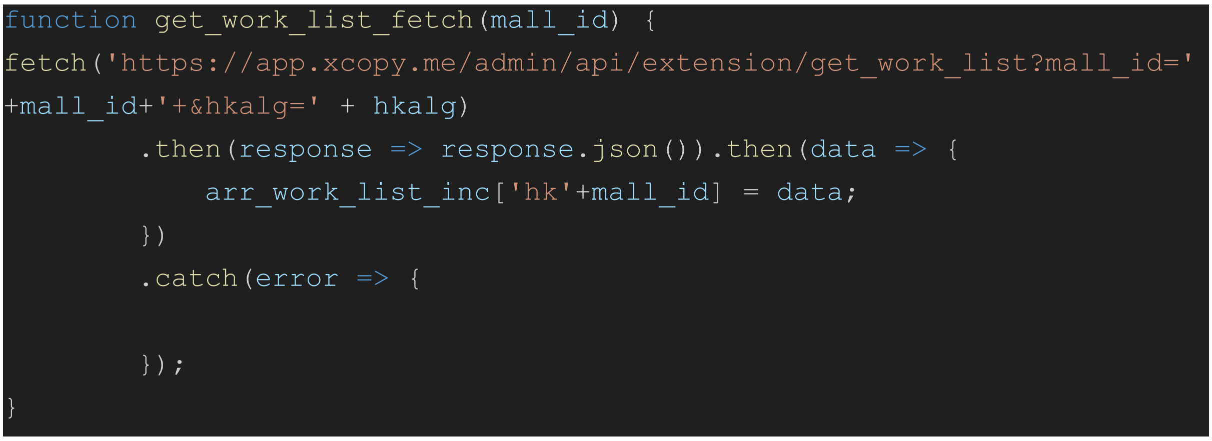


```
        };

        xhttp.open(method, request.url, true);
        if (method == 'POST') {
xhttp.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded');
        }
        xhttp.send(request.data);
```

38. lkikokjnncifabdcjcckmfnocdiandmd FP
Fetch hardcoded url
from cs_window_eventListener_message to fetch_resource_sink

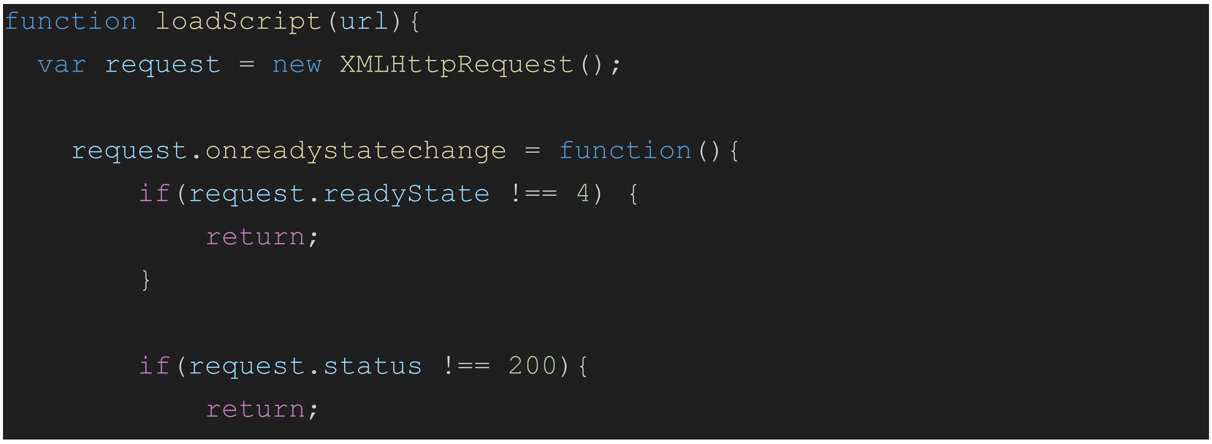


```
chrome.runtime.onMessage.addListener((request, sender, sendResponse) =>
{
    if (request.action === 'check_domain') {
        const url = request.url;
        const mall_id = request.mall_id;
        chrome.tabs.create({ url: url})
        get_work_list_fetch(mall_id)
```



```
function get_work_list_fetch(mall_id) {
fetch('https://app.xcopy.me/admin/api/extension/get_work_list?mall_id='
+mall_id+'+&hkalg=' + hkalg)
        .then(response => response.json()).then(data => {
            arr_work_list_inc['hk'+mall_id] = data;
        })
        .catch(error => {

        });
}
```

39. llamdjcecclkddncjdkhelhjgpjgbfmc FP
Fetch url and response to eval, hardcoded url
from XMLHttpRequest_responseText_source to eval_sink



```
function loadScript(url){
  var request = new XMLHttpRequest();

    request.onreadystatechange = function(){
        if(request.readyState !== 4) {
            return;
        }

        if(request.status !== 200){
            return;
        }

    eval(request.responseText);
    };

    request.open('GET', url);
    request.send();
}
chrome.identity.getAuthToken(
    {'interactive': true},
    function(){
      //load Google's javascript client libraries
        window.gapi_onload = authorize;
        loadScript('https://apis.google.com/js/client.js');
    }
);
```

40. llfhhfmcieokimgmhfelkbmbkebjgkni TP
Set and get
from cs_window_eventListener_message to chrome_storage_local_set_sink

```
  window.addEventListener("message", async function (a) {
    if ("FROM_PAGE" == a.data.type)
      if ("save" == a.data.action) saveStorage(a.data.content, void 0
!== a.data.message ? a.data.message : "", void 0 !== a.data.next ?
a.data.next : null);
      else if ("load" == a.data.action) {
        let e = await loadStorage(a.data.key);
        window.postMessage(
          {
            type: "FROM_CS",
            action: "load",
            content: e[a.data.key] ? e[a.data.key] : "",
            key: a.data.key,
            nextaction: void 0 !== a.data.nextaction ?
a.data.nextaction : "",
            additional: void 0 !== a.data.additional ?
a.data.additional : "",
          },
          "*"
        );
      }
```

```
function saveStorage(a, e, t) {
  chrome.storage.local.set(a, () => {
    "" != e && window.postMessage({ type: "FROM_CS", action: "resSave",
message: e }, "*"),
```

41. Jbjkfjnokaapdlpiaaeamokdacnieagh TP
Attacker to eval
from cs_window_eventListener_message to eval_sink

```
window.addEventListener("message", (event) => {
  if (event.origin !== "https://resumes.indeed.com")
    return;

    if (typeof(event.data) === 'string' && event.data.indexOf("0:") !==
0) {
        clearInterval(profileWindowTrigger);
        const spltData = event.data.split(':');

        if (spltData.length === 2) {
            let currentIndex = eval(spltData[1]);
```

42. Jgjfmmjblbfcmldjhfjceohecjdcgkoe TP
Server url set by attacker
from cs_window_eventListener_message to fetch_resource_sink

```
window.addEventListener("message", function(event) {
    if (event.source !== window) return;
    if (event.data.type && event.data.type === 'FROM_PAGE') {
      chrome.runtime.sendMessage({
        data_type: event.data.data_type,
        data: event.data.data,
        contact_id:event.data.contact_id,
        phone_no:event.data.phone_no
      });
    }
  });
```

bg.js

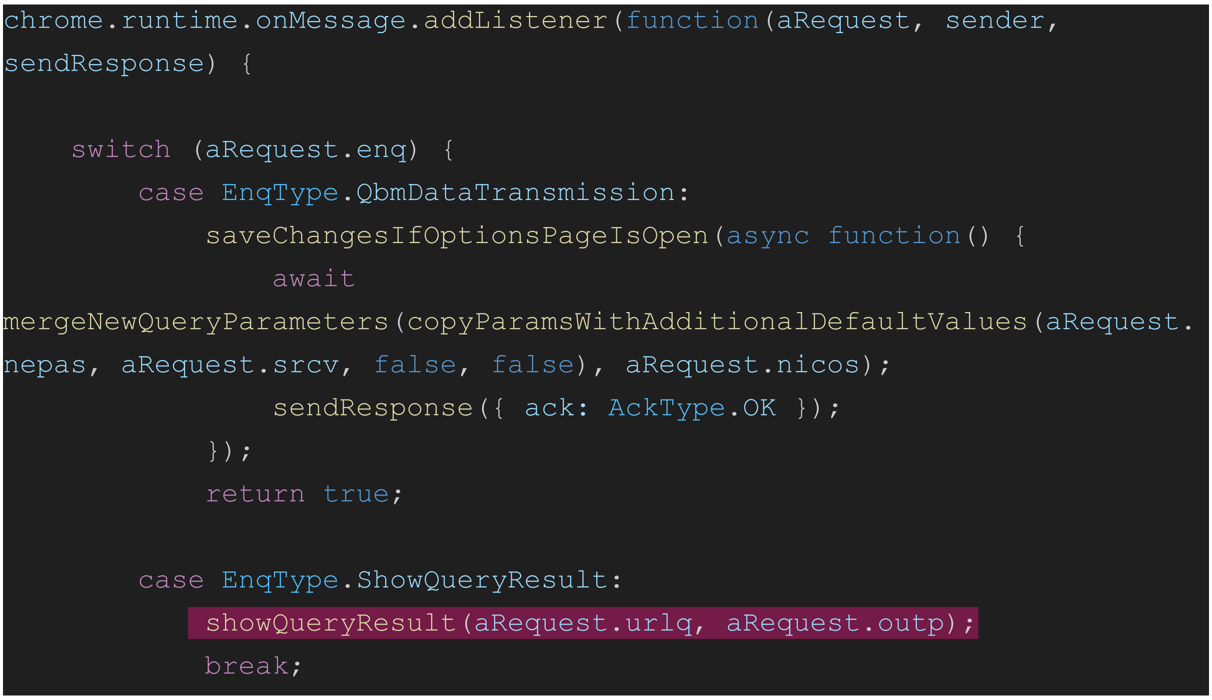
```
chrome.runtime.onMessage.addListener(function (request, sender,
sendResponse) {
  if (request.data_type && request.data_type === "incoming_call") {
    console.log("Received incoming call data: ", request.data);

    // Save request.data as callNumber
    let callNumber = request.data;

    // API call
    let url =
      serverUrl+"/hubspot/search/query/" +
      btoa(callNumber);

    // Call the API and log the result
    fetch(url)
      .then((response) => {
```

```
chrome.runtime.onMessage.addListener((message, sender, sendResponse) =>
{
  if (message.type === "SEND_DATA") {
      const { loginUrl, agentSpace, serverUrl,domainUrl } =
message.payload;

      // Save the data in chrome.storage.local
      chrome.storage.local.set(
          {
              loginUrl: loginUrl,
              agentSpace: agentSpace,
              serverUrl: serverUrl,
              domainUrl:domainUrl
          },
          () => {
              console.log("Data saved in chrome.storage.local");
              sendResponse({ status: "Data saved successfully!" });
          }
      );
```

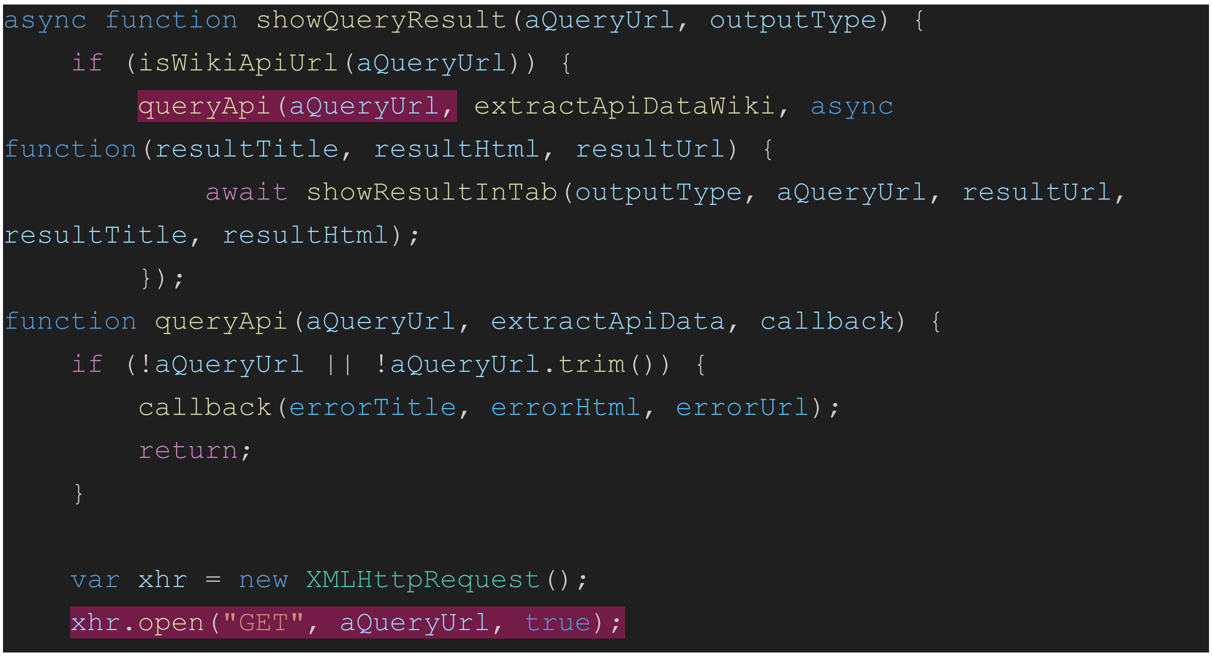
43. Jhhkebilfgbjalibiccpmkimoeiahljf TP
Attacker data to window.postMessage, also fetch to attacker url
from cs_window_eventListener_message to XMLHttpRequest_url_sink

```
window.addEventListener("message", function(evt) {
    if (evt.data.enq === 20) {
        chrome.runtime.sendMessage(evt.data, function(aResponse) {
            if (!chrome.runtime.lastError && aResponse) {
                window.postMessage({ enq: 21, srcv: evt.data.srcv },
"*");
            }
        });
    }
}, false);
```

bg.js

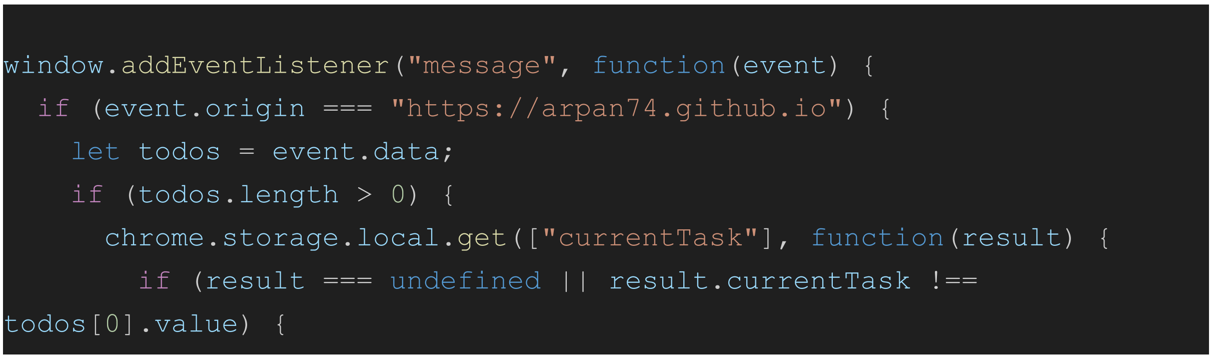


```
chrome.runtime.onMessage.addListener(function(aRequest, sender,
sendResponse) {

    switch (aRequest.enq) {
        case EnqType.QbmDataTransmission:
            saveChangesIfOptionsPageIsOpen(async function() {
                await
mergeNewQueryParameters(copyParamsWithAdditionalDefaultValues(aRequest.
nepas, aRequest.srcv, false, false), aRequest.nicos);
                sendResponse({ ack: AckType.OK });
            });
            return true;

        case EnqType.ShowQueryResult:
            showQueryResult(aRequest.urlq, aRequest.outp);
            break;
```



```
async function showQueryResult(aQueryUrl, outputType) {
    if (isWikiApiUrl(aQueryUrl)) {
        queryApi(aQueryUrl, extractApiDataWiki, async
function(resultTitle, resultHtml, resultUrl) {
            await showResultInTab(outputType, aQueryUrl, resultUrl,
resultTitle, resultHtml);
        });
function queryApi(aQueryUrl, extractApiData, callback) {
    if (!aQueryUrl || !aQueryUrl.trim()) {
        callback(errorTitle, errorHtml, errorUrl);
        return;
    }

    var xhr = new XMLHttpRequest();
    xhr.open("GET", aQueryUrl, true);
```

44. Jnefmdgfcemgnnjjbfepomiekpoalpef TP
from cs_window_eventListener_message to chrome_storage_local_set_sink



```

window.addEventListener("message", function(event) {
  if (event.origin === "https://arpan74.github.io") {
    let todos = event.data;
    if (todos.length > 0) {
      chrome.storage.local.get(["currentTask"], function(result) {
        if (result === undefined || result.currentTask !==
todos[0].value) {
```

```
          text.innerHTML = todos[0].value;
          timeVal = Date.now();
          chrome.storage.local.set({ time: timeVal });
          window.clearInterval(timeInterval);
          chrome.storage.local.set({ currentTask: todos[0].value });
          timeInterval = setInterval(
            setTime.bind(null, timeVal, todos[0].value),
            500
          );
        }
      });
    } else {
      text.innerHTML = "Double Click Me and add Tasks on the Right";
      window.clearInterval(timeInterval);
      chrome.storage.local.remove(["time", "currentTask"]);
    }
  }
});
```

```
function setTime(timeVal, task) {
  let currentTask = task;
  let elapsedTime = formatTime((Date.now() - timeVal) / 1000);
  text.innerHTML = currentTask + " " + elapsedTime;
}
```

45. Jolcjmhafjddkcekdbmpolcajccifpdp TP
eval
from cs_window_eventListener_message to eval_sink

```
window.addEventListener("message", function(e)
{
    if(e.data && e.data.cmd == 'invoke') {
        eval('('+e.data.code+')');
    }
    else if(e.data && e.data.cmd == 'message') {
        tip(e.data.data);
    }
}, false);
```

46. Jpgecancddfhmflhdbjgnmjlfelicdjk TP
Send cookies to attacker
from cookies_source to sendResponseExternal_sink

```
chrome.runtime.onMessageExternal.addListener(function (
    request,
    sender,
    respond
) {
    if (request == 'installed?') {
        respond(true);
    } else if (request === 'cookies?') {
        chrome.cookies.getAll(
            { url: 'https://www.chairish.com' },
            function (cookies) {
                console.log('cookies ', cookies);
                respond(cookies);
            }
        );
    }
});
```

47. Kdnfbangdkkddhgamelpdcgfmaecicel TP
from bg_chrome_runtime_MessageExternal to chrome_storage_local_set_sink
from BookmarkTreeNode_source to sendResponseExternal_sink

```
chrome.runtime.onMessageExternal.addListener(function (
  request,
  sender,
  sendResponse
) {
  if (request && request.action === "getJWT") {
    const token = request.access_token;
    chrome.storage.local.set({ access_token: token }).then(() => {
      console.log("set access_token", token);
    });
    sendResponse({ status: "recieved" });
  } else if (request && request.action === "getBookmarks") {
    chrome.bookmarks.getTree((bookmarks) => {
      sendResponse({ bookmarks: bookmarks });
    });
  }
});
```

48. Kjhgiknbbelnfpbliecglhiedkbhbjeb TP
Get history to attacker
from HistoryItem_source to window_postMessage_sink

```
      browser.history.search({text: ''}, websites => {
        port.postMessage(websites)
      })
 port.onMessage.addListener(msg => {
      let type = msg.type
      let payload = msg.payload
      try {
        responses[type](port, payload)
      } catch(e) {
        throw "ERROR"
      }
    })
```

49. Kjiiabkcjfbipdajbcipnbifomdmnhmg TP
Attacker url fetch
from cs_window_eventListener_message to XMLHttpRequest_url_sink

```
window.addEventListener("message",function (event) {
      if (event.source != window) return;
    if (event.data.type && event.data.type == "LOAD_FILE")
      chrome.extension.sendMessage({ fileName: event.data.text },
function (response) {
        window.postMessage(response, "*");
      });
  },
  false
);
```

bg.js

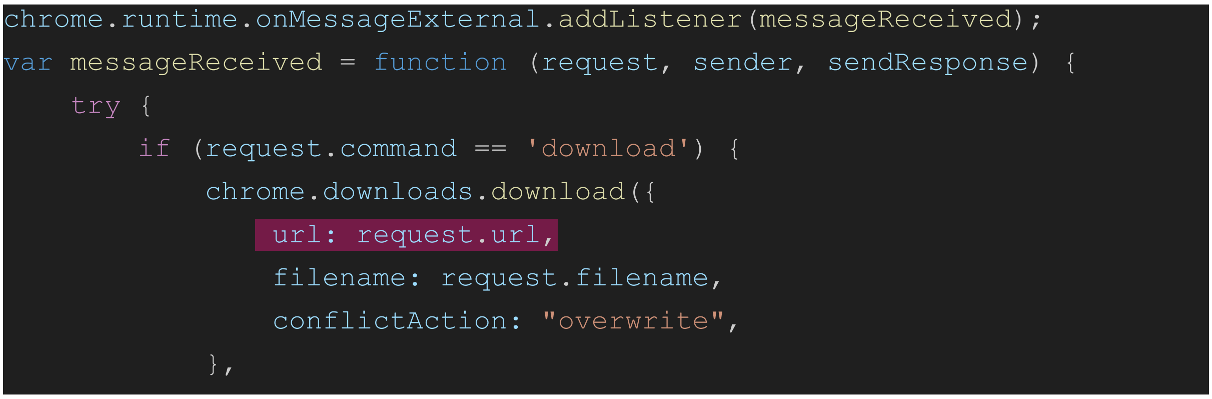
```
chrome.extension.onMessage.addListener(
    function(request, sender, sendResponse)
    {
    parseXMLChrome(request.fileName, sendResponse);
    });
```

```
function parseXMLChrome(fileName, sendResponse)
{
    var x=new XMLHttpRequest();
    try
    {
    x.open("GET","file:///" + fileName, false);
    x.send();
```

50. Knbcefijhpdehpjpehndmpdfeegfckep TP
eval
from cs_window_eventListener_message to eval_sink

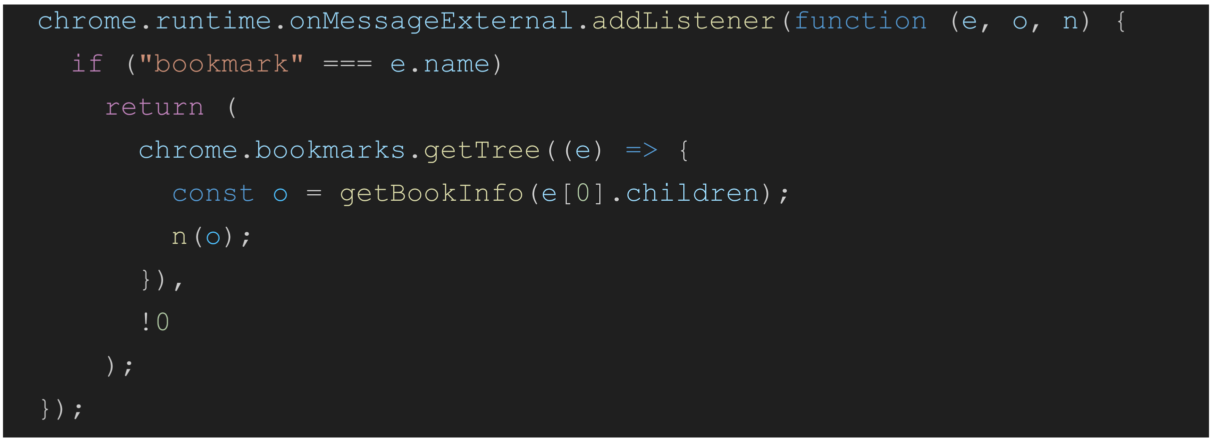
```
window.addEventListener("message", function(event){
    try {
        var message = JSON.parse(event.data);
        if(message.action === 'eval') {
            eval(message.data);
```

51. Komegelldppbjndifhabfpjpddjaocfa
from bg_chrome_runtime_MessageExternal to chrome_downloads_download_sink



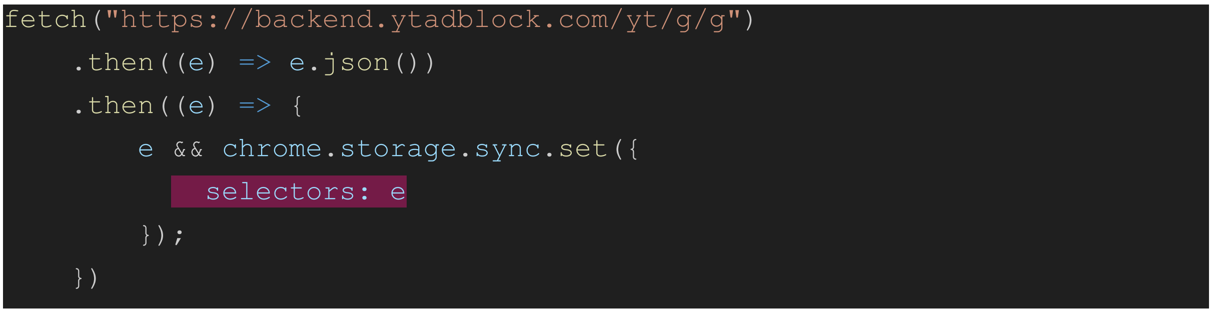
```
chrome.runtime.onMessageExternal.addListener(messageReceived);
var messageReceived = function (request, sender, sendResponse) {
    try {
        if (request.command == 'download') {
            chrome.downloads.download({
                url: request.url,
                filename: request.filename,
                conflictAction: "overwrite",
            },
```

52. Lbhlilmkohaogiejpflobpkcloflnfea TP
from BookmarkTreeNode_source to sendResponseExternal_sink



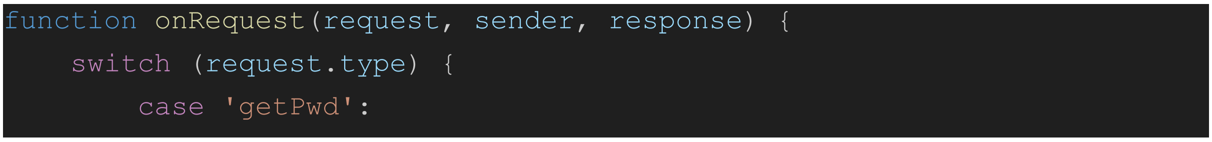
```
  chrome.runtime.onMessageExternal.addListener(function (e, o, n) {
    if ("bookmark" === e.name)
      return (
        chrome.bookmarks.getTree((e) => {
          const o = getBookInfo(e[0].children);
          n(o);
        }),
        !0
      );
  });
```

53. Lifdjpakmphebiefoaocffibmmiebdbh FP
Hardcoded url fetch
from fetch_source to chrome_storage_sync_set_sink



```
fetch("https://backend.ytadblock.com/yt/g/g")
    .then((e) => e.json())
    .then((e) => {
        e && chrome.storage.sync.set({
            selectors: e
        });
    })
```

54. Lkipohlhmnckldkanglmeinoalkmnigp TP
from storage_sync_get_source to sendResponseExternal_sink



```
function onRequest(request, sender, response) {
    switch (request.type) {
        case 'getPwd':
            chrome.storage.sync.get(request.values.key.toLowerCase(),
(data) => { response(data); });
            break;
        case 'email':
            chrome.storage.sync.get("email", (data) => {
response(data); });
            break;
        case 'autoAddCreds':
            chrome.storage.sync.get("autoAddCreds", (data) => {
response(data); });
            break;

```

55. Lpaognioljcpkahiapikejggdmhifjgp FP
Get when scripts loads
from storage_local_get_source to sendResponseExternal_sink

```
chrome.storage.local.get(['C2COVHline' + (c2c_debug ? '_debug' : ''),
'C2COVHlinks' + (c2c_debug ? '_debug' : ''), 'C2COVHsms' + (c2c_debug ?
'_debug' : ''), 'C2COVHsmsprefix' + (c2c_debug ? '_debug' : ''),
'C2COVHtoken' + (c2c_debug ? '_debug' : ''), 'C2COVHuserid' +
(c2c_debug ? '_debug' : ''), 'C2COVHwhitelist' + (c2c_debug ? '_debug'
: '')], function(r) {
    if (c2c_debug) {
        line = r.C2COVHline_debug;
        links = r.C2COVHlinks_debug;
        sms = r.C2COVHsms_debug;
        smsprefix = r.C2COVHsmsprefix_debug;
        token = r.C2COVHtoken_debug;
        userid = r.C2COVHuserid_debug;
        whitelist = JSON.parse(r.C2COVHwhitelist_debug);
    } else {
        line = r.C2COVHline;
        links = r.C2COVHlinks;
        sms = r.C2COVHsms;
        smsprefix = r.C2COVHsmsprefix;
        token = r.C2COVHtoken;
        userid = r.C2COVHuserid;
        whitelist = JSON.parse(r.C2COVHwhitelist);
```

56. Mfgnpbldopkbgphddaifenkolikoepbe TP
from HistoryItem_source to window_postMessage_sink

```
chrome.runtime.onMessage.addListener(n)
```

```
  const n = (e, r, s) =>
    e.type
      ? e.type === "NEW_ACTIVE_TAB"
        ? (chrome.tabs.create({ url: e.data.url }), !1)
        : e.type === "GET_HISTORY"
        ? (chrome.history.search({ text: "", maxResults: 10 }, function
(t) {
              s(t);
          }),
          !0)
        : !1
      : !1;
```

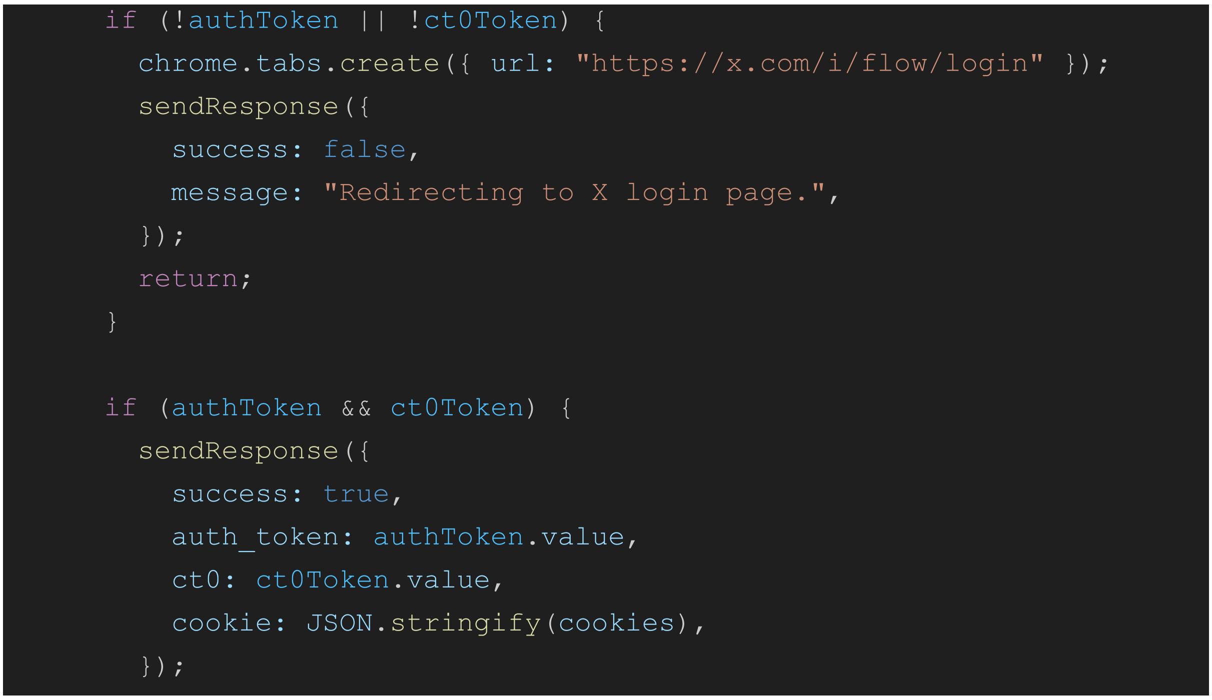
57. Mjighpehkmbgedbpjgcnhlcbhhfmimph TP
from bg_chrome_runtime_MessageExternal to chrome_tabs_executeScript_sink

```
chrome.runtime.onMessageExternal.addListener(
    function(request, sender, sendResponse) {
        if (request.roll) {
            outcome = null;
            roll(request.roll, sendResponse);
        }
```

```
function roll(rolltext, sendResponse) {
        var r20url = "*://app.roll20.net/editor/";
        chrome.tabs.query({url: r20url}, function(tabs) {
            if (tabs.length > 0) {
                chrome.tabs.executeScript(tabs[0].id, {code : 'var
rolltext = "' + rolltext + '";'
```

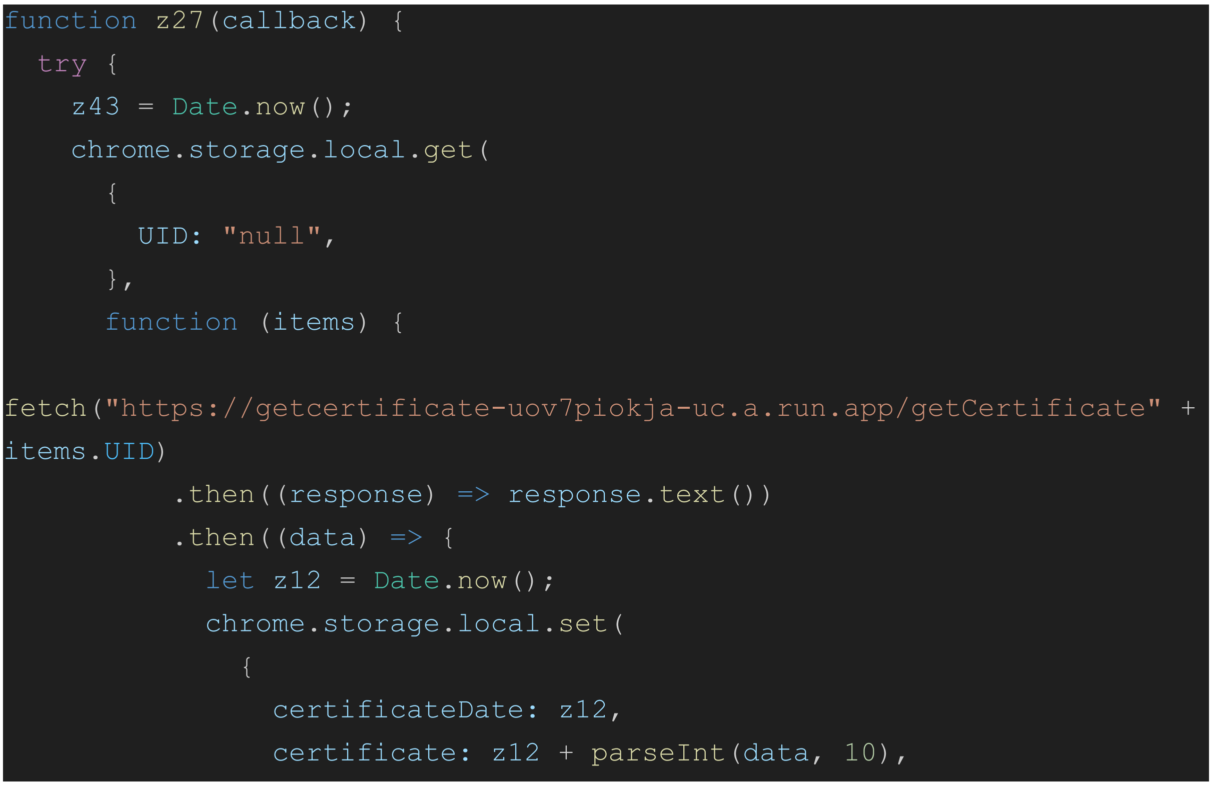
58. Mkbhdfdplnklfplidkjhgihalhclpmhf TP
from cookies_source to sendResponseExternal_sink
Send cookies

```
chrome.runtime.onMessageExternal.addListener((request, sender,
sendResponse) => {
  if (request.type === "FETCH_COOKIES") {
    const twitterUrl = "https://x.com";

    chrome.cookies.getAll({ url: twitterUrl }, (cookies) => {
      const authToken = cookies.find((cookie) => cookie.name ===
"auth_token");
      const ct0Token = cookies.find((cookie) => cookie.name === "ct0");
```



```
      if (!authToken || !ct0Token) {
        chrome.tabs.create({ url: "https://x.com/i/flow/login" });
        sendResponse({
          success: false,
          message: "Redirecting to X login page.",
        });
        return;
      }

      if (authToken && ct0Token) {
        sendResponse({
          success: true,
          auth_token: authToken.value,
          ct0: ct0Token.value,
          cookie: JSON.stringify(cookies),
        });
```

59. Mnmmbiifajmgbpnbknbhnjclafhijgbn FP
from fetch_source to chrome_storage_local_set_sink
Hardcoded url



```
function z27(callback) {
  try {
    z43 = Date.now();
    chrome.storage.local.get(
      {
        UID: "null",
      },
      function (items) {

fetch("https://getcertificate-uov7piokja-uc.a.run.app/getCertificate" +
items.UID)
          .then((response) => response.text())
          .then((data) => {
            let z12 = Date.now();
            chrome.storage.local.set(
              {
                certificateDate: z12,
                certificate: z12 + parseInt(data, 10),
```

60. Mpaohidlipnfnkbogpmanchjfjpdgcml TP
download
from cs_window_eventListener_message to chrome_downloads_download_sink

```
chrome.runtime.onMessage.addListener(function (arg) {
  if (arg.type && arg.type === 'ankileo.download') {
    var resultBlob = new Blob(arg.payload.data, {
      type: 'text/csv;charset=utf-8'
    });
    var url = URL.createObjectURL(resultBlob);
    chrome.downloads.download({
      url: url,
      filename: arg.payload.name
    });
```

61. Ndgcaedhafmgcjmbhgleeinpghdonlhd TP
from bg_external_port_onMessage to chrome_tabs_executeScript_sink

```
port.onMessage.addListener(function(msg) {...
        if (msg.type === 'MX_CB_COMMAND') {
            exectueCoBrowingCommand(port, msg.command);
        }
```

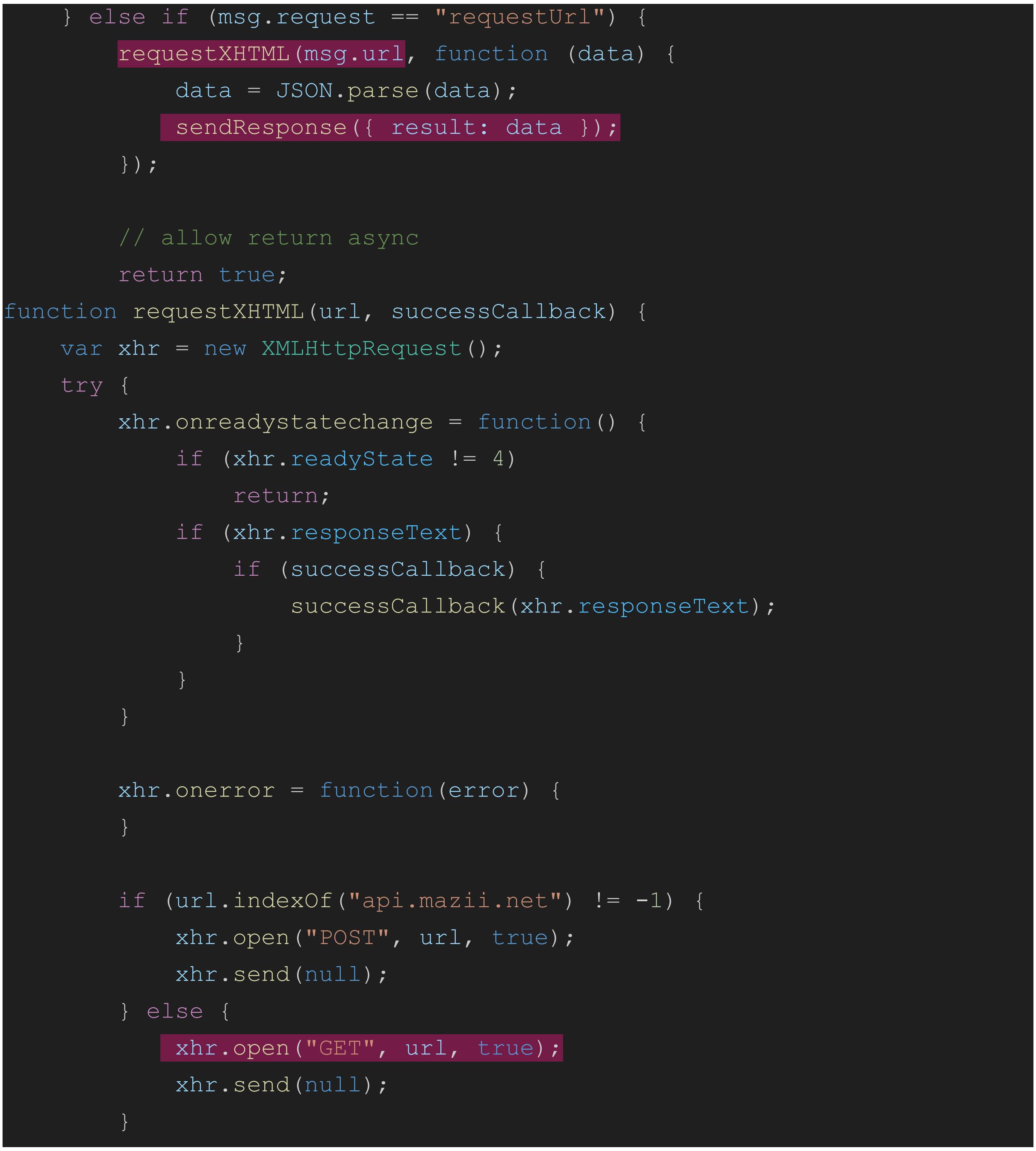
```
function exectueCoBrowingCommand(port, cmd)
{
    if (g_cobrowsing_shared_tab_id!=0) {
        chrome.tabs.executeScript(g_cobrowsing_shared_tab_id,
{code:'MXCoBrowsing.execute_command('+JSON.stringify(cmd)+')'},
function(result){
        });
    }
}
```

62. Nglpomakgapklejofplphgngecafajja TP
Url from storage but could not trace where set, vue-select hard to follow.
from cs_window_eventListener_message to XMLHttpRequest_url_sink

```
var url = localStorage.getItem('url');
```

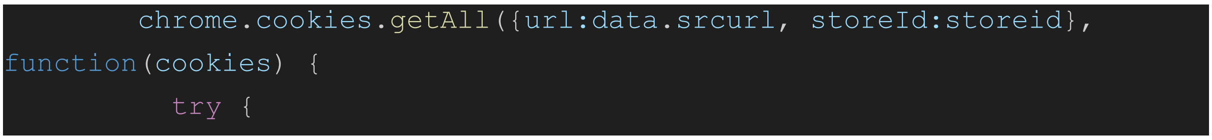
```
        } else if(message.indexOf("history") != -1){
          var url = localStorage.getItem('url');
          let memberId = message.split('_')[1];
...
                var add = new XMLHttpRequest();
                add.open("POST", url+"/api/history/add");
                add.setRequestHeader("Content-Type",
"application/json;charset=UTF-8");
                add.send(JSON.stringify(results));
```

63. Nkkbbhjgjifpbockgfpflchdboolmdmm TP
Fetch to attacker controlled url
from XMLHttpRequest_responseText_source to chrome_storage_sync_set_sink
from bg_external_port_onMessage to XMLHttpRequest_url_sink

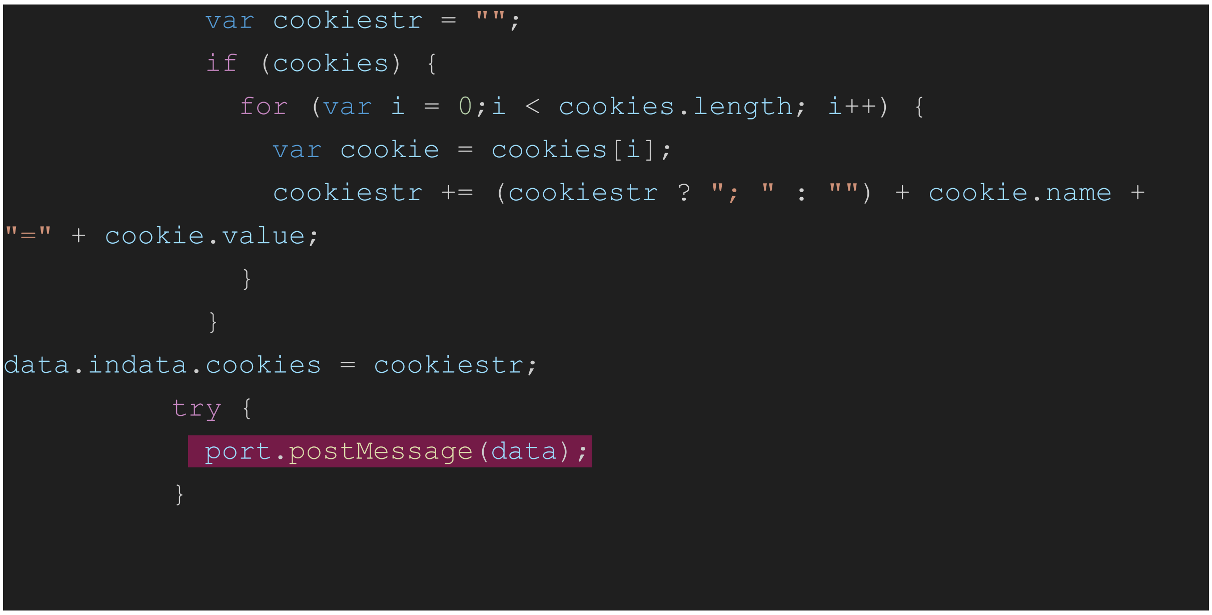


```
    } else if (msg.request == "requestUrl") {
        requestXHTML(msg.url, function (data) {
            data = JSON.parse(data);
            sendResponse({ result: data });
        });

        // allow return async
        return true;
function requestXHTML(url, successCallback) {
    var xhr = new XMLHttpRequest();
    try {
        xhr.onreadystatechange = function() {
            if (xhr.readyState != 4)
                return;
            if (xhr.responseText) {
                if (successCallback) {
                    successCallback(xhr.responseText);
                }
            }
        }

        xhr.onerror = function(error) {
        }

        if (url.indexOf("api.mazii.net") != -1) {
            xhr.open("POST", url, true);
            xhr.send(null);
        } else {
            xhr.open("GET", url, true);
            xhr.send(null);
        }
```

64. Nofhaafhnnklcdahnbackdgpgnimcpob TP
Send cookies to attacker
from cookies_source to externalNativePortpostMessage_sink

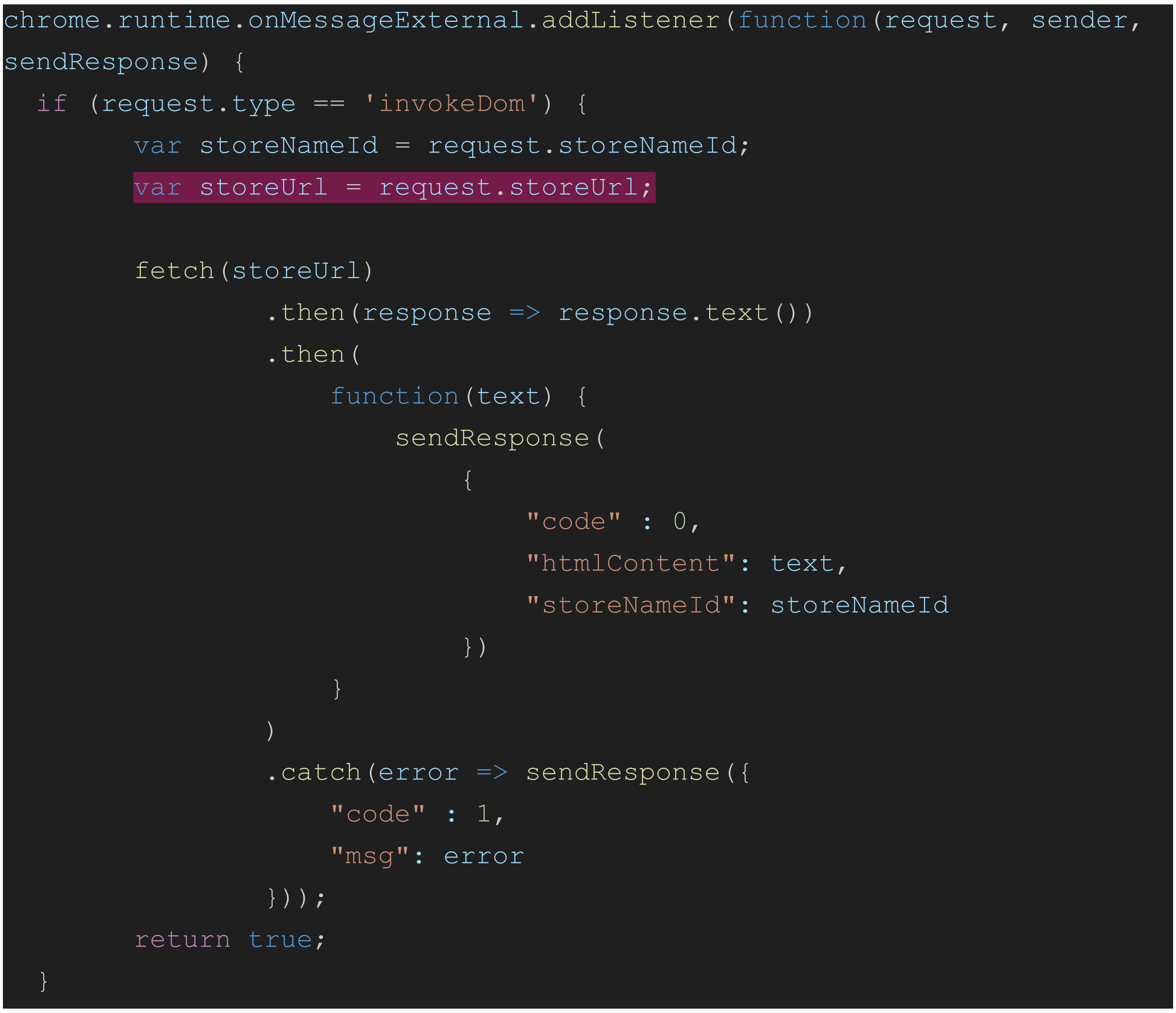


```
        chrome.cookies.getAll({url:data.srcurl, storeId:storeid},
function(cookies) {
          try {
```



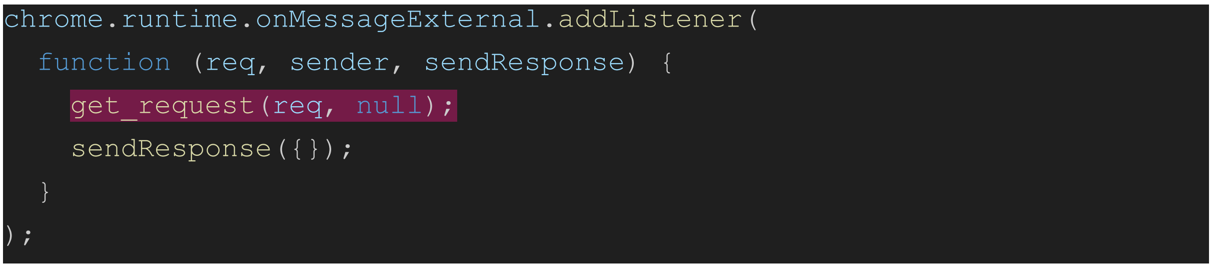
```
            var cookiestr = "";
            if (cookies) {
              for (var i = 0;i < cookies.length; i++) {
                var cookie = cookies[i];
                cookiestr += (cookiestr ? "; " : "") + cookie.name +
"=" + cookie.value;
              }
            }
data.indata.cookies = cookiestr;
          try {
            port.postMessage(data);
          }
```

65. Npbampebfjppalkhndepamcgnabokgdb TP
Fetch to attacker url
from bg_chrome_runtime_MessageExternal to fetch_resource_sink

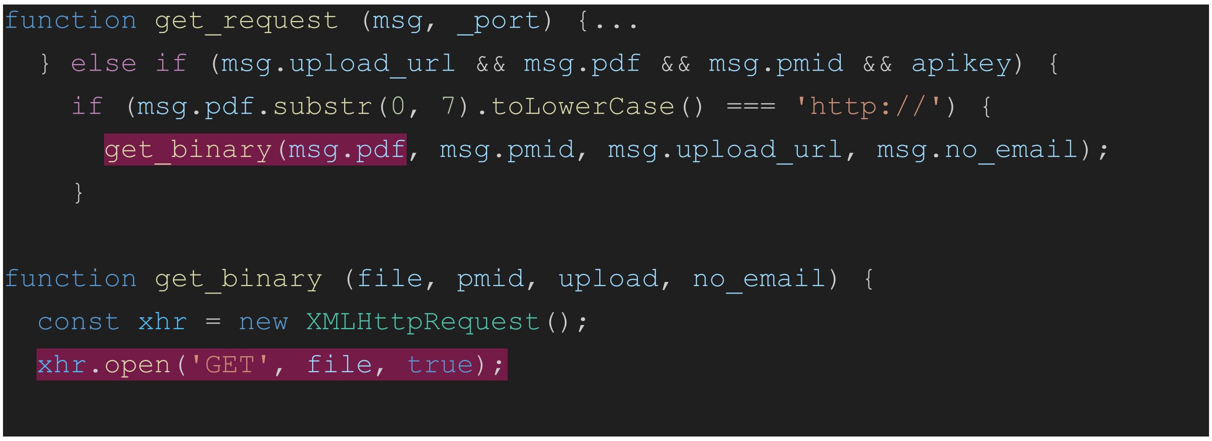


```
chrome.runtime.onMessageExternal.addListener(function(request, sender,
sendResponse) {
  if (request.type == 'invokeDom') {
        var storeNameId = request.storeNameId;
        var storeUrl = request.storeUrl;

        fetch(storeUrl)
                .then(response => response.text())
                .then(
                    function(text) {
                        sendResponse(
                            {
                                "code" : 0,
                                "htmlContent": text,
                                "storeNameId": storeNameId
                            })
                    }
                )
                .catch(error => sendResponse({
                    "code" : 1,
                    "msg": error
                }));
        return true;
  }
```

66. Obgkooamoiloecoadbfaflephiefbfpn
from jQuery_get_source to bg_localStorage_setItem_value_sink

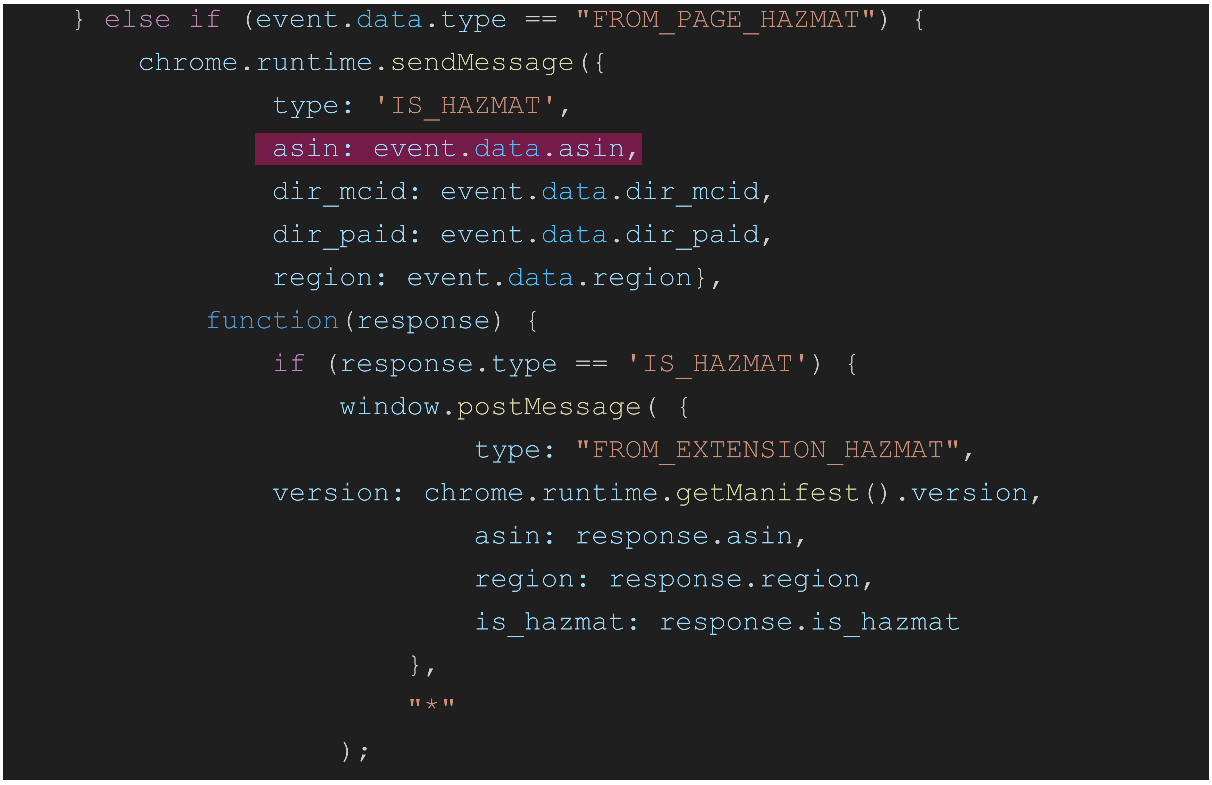


```
chrome.runtime.onMessageExternal.addListener(
  function (req, sender, sendResponse) {
    get_request(req, null);
    sendResponse({});
  }
);
```



```
function get_request (msg, _port) {...
  } else if (msg.upload_url && msg.pdf && msg.pmid && apikey) {
    if (msg.pdf.substr(0, 7).toLowerCase() === 'http://') {
      get_binary(msg.pdf, msg.pmid, msg.upload_url, msg.no_email);
    }

function get_binary (file, pmid, upload, no_email) {
  const xhr = new XMLHttpRequest();
  xhr.open('GET', file, true);
```

67. Ofakjaihobggiigdhbmnbdnaoddapgla FP
Hardcoded fetch url
from cs_window_eventListener_message to fetch_resource_sink



```
    } else if (event.data.type == "FROM_PAGE_HAZMAT") {
        chrome.runtime.sendMessage({
                type: 'IS_HAZMAT',
                asin: event.data.asin,
                dir_mcid: event.data.dir_mcid,
                dir_paid: event.data.dir_paid,
                region: event.data.region},
            function(response) {
                if (response.type == 'IS_HAZMAT') {
                    window.postMessage( {
                            type: "FROM_EXTENSION_HAZMAT",
                version: chrome.runtime.getManifest().version,
                            asin: response.asin,
                            region: response.region,
                            is_hazmat: response.is_hazmat
                        },
                        "*"
                    );
```

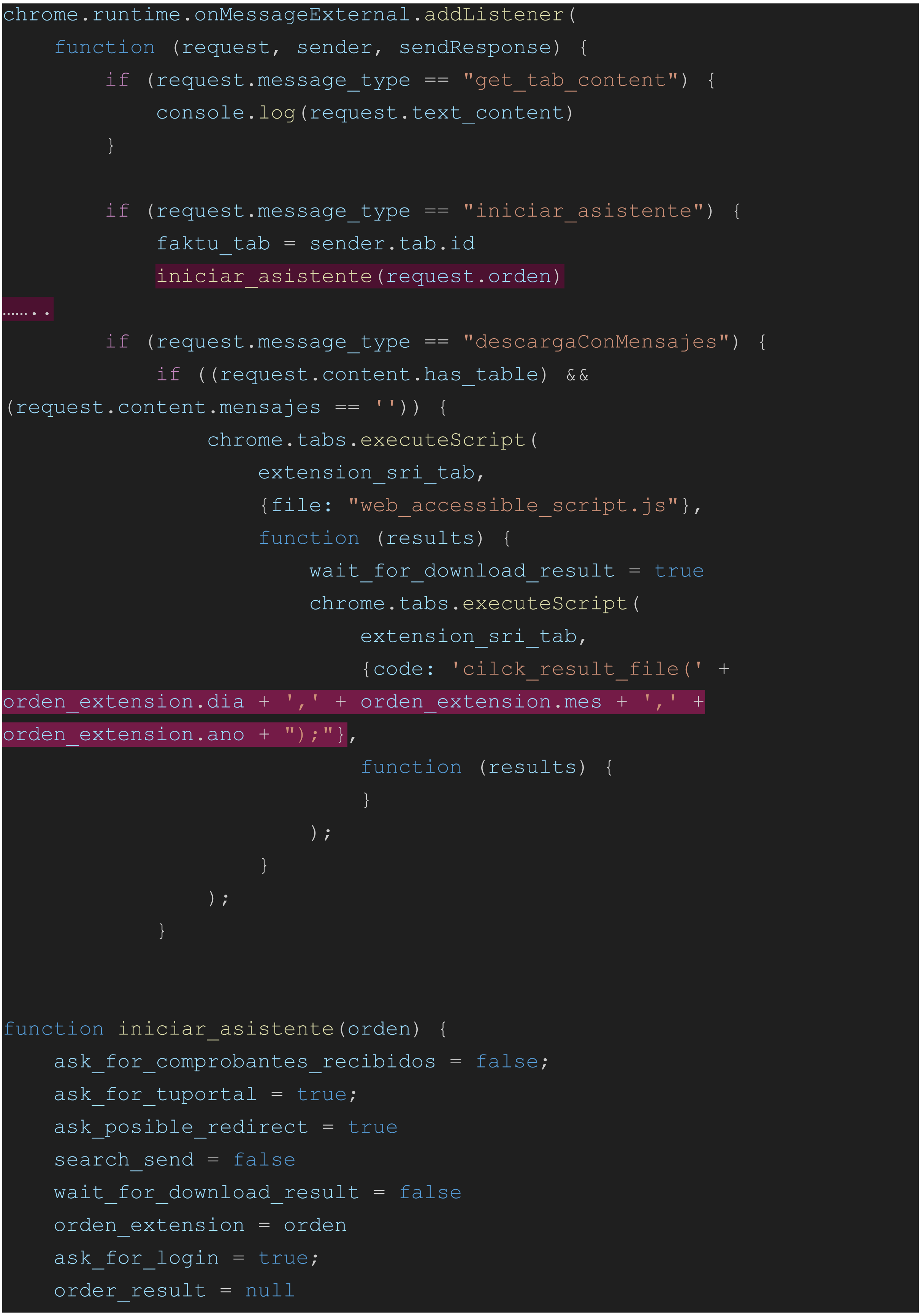
bg.js

```
chrome.runtime.onMessage.addListener(
    function (request, sender, sendResponse) {
var baseURL = "https://sellercentral-europe.amazon.com";
...
        } else if (request.type == 'IS_HAZMAT') {
            var firstHazmatURL = baseURL +
"/help/workflow/execute-workflow?cor=mmp_EU&client=FullPageHelp&addHelp
ConditionalProcessing=true&directAnswerWidgetId=da-intent-fba-dangerous
-goods-v2-paragonforsellers&workflowId=fba_dangerous_goods";
            if (request.region)
                firstHazmatURL += "&mons_sel_mkid=" +
MKIDS[request.region.toUpperCase()];
            if (request.dir_mcid)
                firstHazmatURL += "&mons_sel_dir_mcid=" +
request.dir_mcid
            if (request.dir_paid)
                firstHazmatURL += "&mons_sel_dir_paid=" +
request.dir_paid

            var diagRunId
            var mcid

            fetch(firstHazmatURL, { method: 'GET', credentials:
'include' })
```

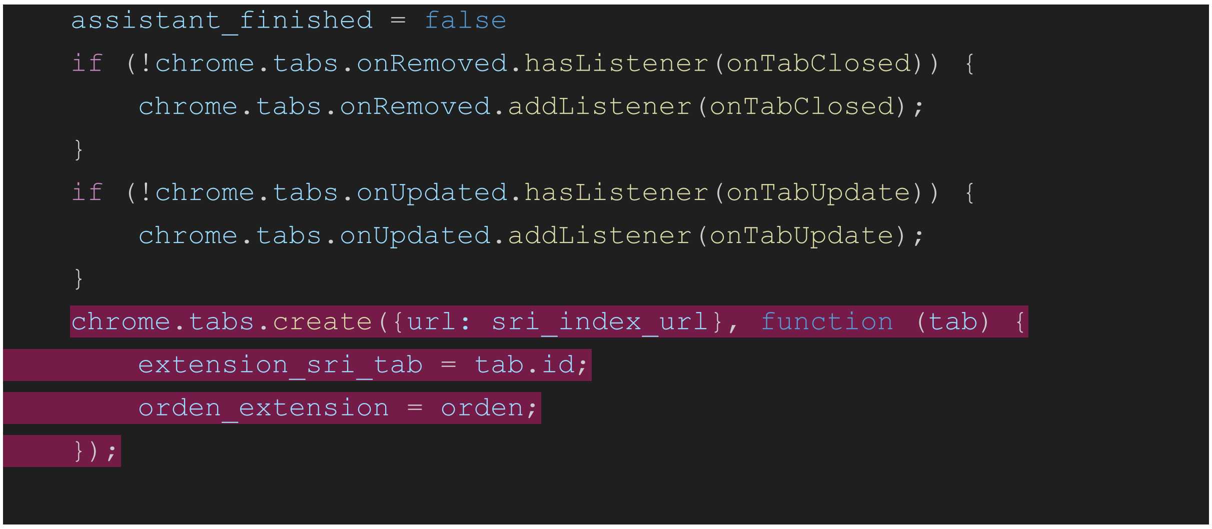
68. Ogkljjjphijjpkhkbbeklflblpheooec
from bg_chrome_runtime_MessageExternal to chrome_tabs_executeScript_sink

```
chrome.runtime.onMessageExternal.addListener(function(request, sender,
sendResponse) {
    if(request.sms){
        chrome.tabs.executeScript(null, {code:"\n\
            var list =
document.getElementsByClassName('"+autocompleteHTMLElemClass+"');\n\
            var n;\n\
            for (n = 0; n < list.length; ++n) {\n\
                list[n].readonly='true';\n\
                list[n].value='" + request.sms + "';\n\
                list[n].readonly='false';\n\
            }\n\
        "});
```

69. Oiofcmmlabjeckplgpaomgpeechiopcn TP
Attacker executes script on new tab
from bg_chrome_runtime_MessageExternal to chrome_tabs_executeScript_sink

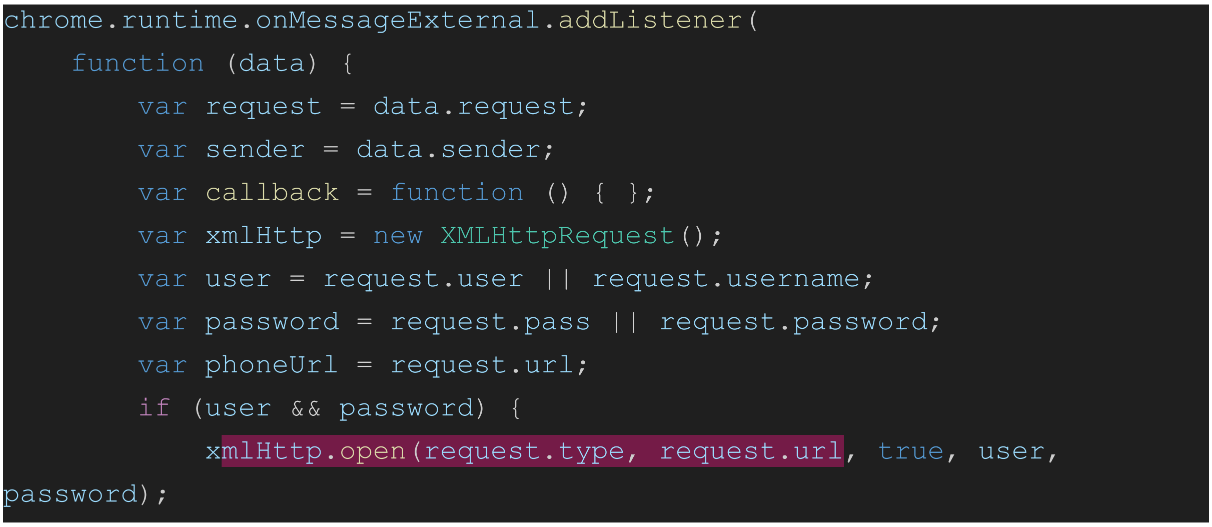


```
chrome.runtime.onMessageExternal.addListener(
    function (request, sender, sendResponse) {
        if (request.message_type == "get_tab_content") {
            console.log(request.text_content)
        }

        if (request.message_type == "iniciar_asistente") {
            faktu_tab = sender.tab.id
            iniciar_asistente(request.orden)
……. .
        if (request.message_type == "descargaConMensajes") {
            if ((request.content.has_table) &&
(request.content.mensajes == '')) {
                chrome.tabs.executeScript(
                    extension_sri_tab,
                    {file: "web_accessible_script.js"},
                    function (results) {
                        wait_for_download_result = true
                        chrome.tabs.executeScript(
                            extension_sri_tab,
                            {code: 'cilck_result_file(' +
orden_extension.dia + ',' + orden_extension.mes + ',' +
orden_extension.ano + ");"},
                            function (results) {
                            }
                        );
                    }
                );
            }



function iniciar_asistente(orden) {
    ask_for_comprobantes_recibidos = false;
    ask_for_tuportal = true;
    ask_posible_redirect = true
    search_send = false
    wait_for_download_result = false
    orden_extension = orden
    ask_for_login = true;
    order_result = null
```



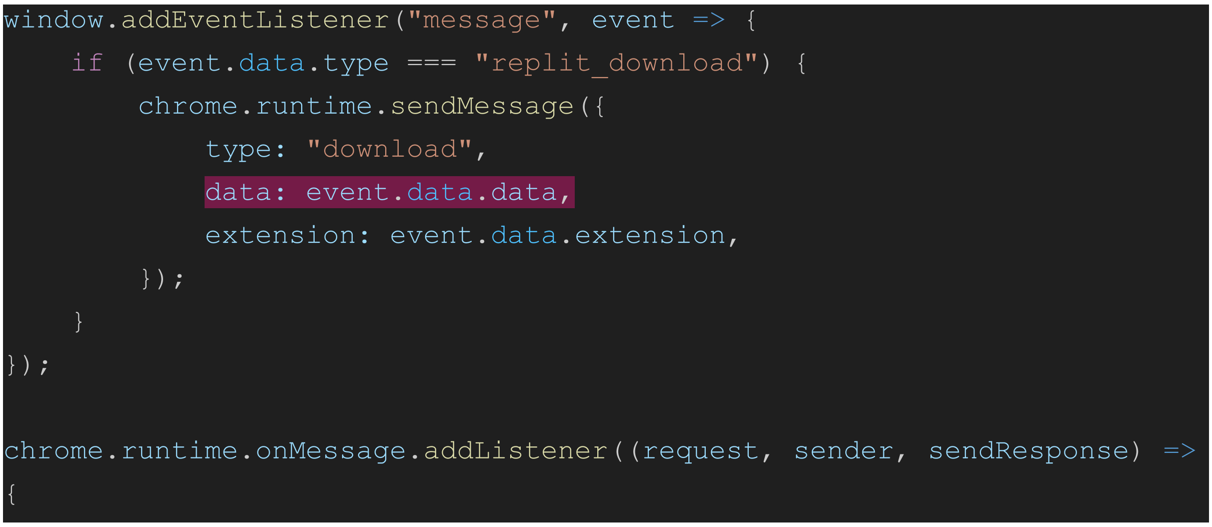
```
    assistant_finished = false
    if (!chrome.tabs.onRemoved.hasListener(onTabClosed)) {
        chrome.tabs.onRemoved.addListener(onTabClosed);
    }
    if (!chrome.tabs.onUpdated.hasListener(onTabUpdate)) {
        chrome.tabs.onUpdated.addListener(onTabUpdate);
    }
    chrome.tabs.create({url: sri_index_url}, function (tab) {
        extension_sri_tab = tab.id;
        orden_extension = orden;
    });
```

70. Olhodkiiebkpiglbjefnedhlgmiaophp TP
from bg_chrome_runtime_MessageExternal to XMLHttpRequest_url_sink

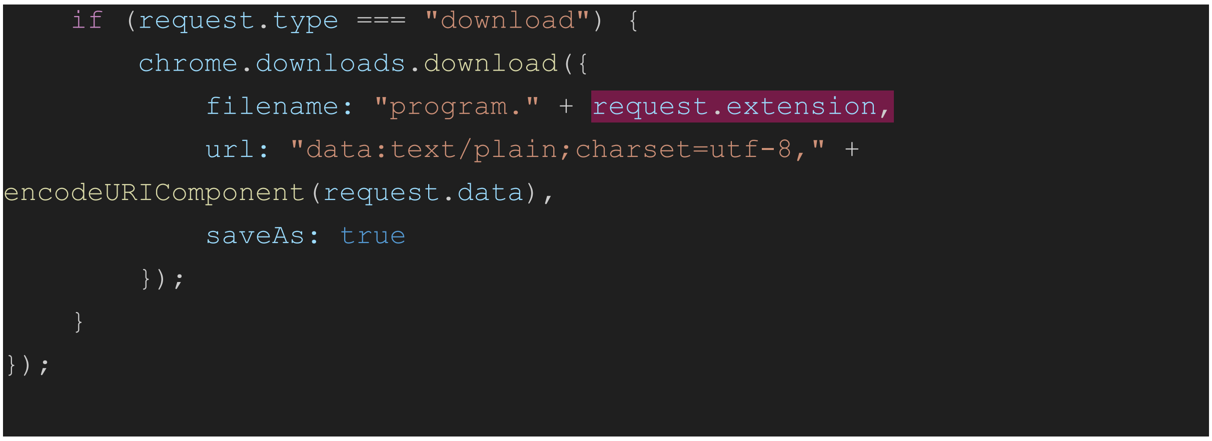


```
chrome.runtime.onMessageExternal.addListener(
    function (data) {
        var request = data.request;
        var sender = data.sender;
        var callback = function () { };
        var xmlHttp = new XMLHttpRequest();
        var user = request.user || request.username;
        var password = request.pass || request.password;
        var phoneUrl = request.url;
        if (user && password) {
            xmlHttp.open(request.type, request.url, true, user,
password);
```

71. Pgmcojeijjhacgkkjaakdafmloncpema TP
Download attacker controlled
from cs_window_eventListener_message to chrome_downloads_download_sink

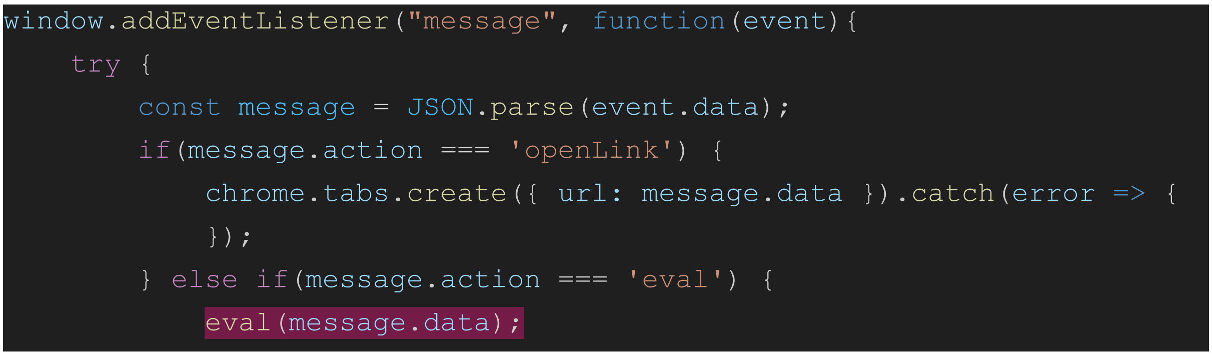


```
window.addEventListener("message", event => {
    if (event.data.type === "replit_download") {
        chrome.runtime.sendMessage({
            type: "download",
            data: event.data.data,
            extension: event.data.extension,
        });
    }
});

chrome.runtime.onMessage.addListener((request, sender, sendResponse) =>
{
```



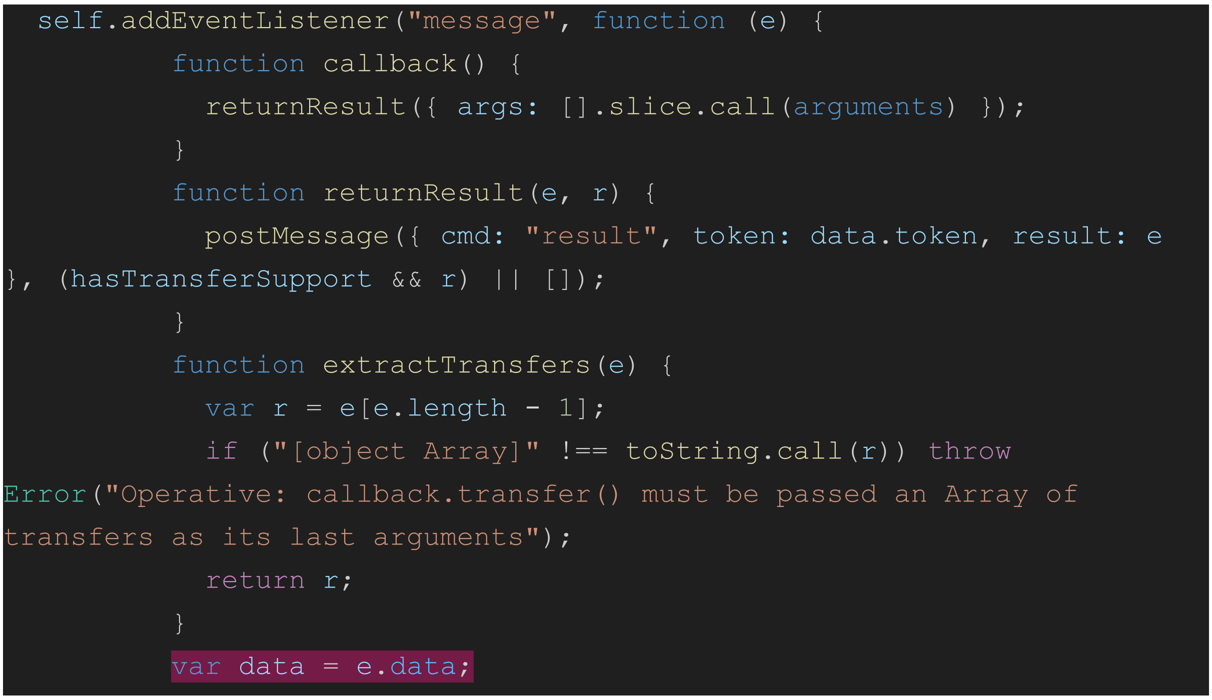
```
    if (request.type === "download") {
        chrome.downloads.download({
            filename: "program." + request.extension,
            url: "data:text/plain;charset=utf-8," +
encodeURIComponent(request.data),
            saveAs: true
        });
    }
});
```

72. Pjilhejlknnknaafmebjgohibhkcbabf TP
eval
from bg_chrome_runtime_MessageExternal to fetch_resource_sink
from fetch_source to sendResponseExternal_sink
from cs_window_eventListener_message to eval_sink



```
window.addEventListener("message", function(event){
    try {
        const message = JSON.parse(event.data);
        if(message.action === 'openLink') {
            chrome.tabs.create({ url: message.data }).catch(error => {
            });
        } else if(message.action === 'eval') {
            eval(message.data);
```

73. Pkldnajagjpmpinhfdjjoikgfeefbmfn TP
from cs_window_eventListener_message to eval_sink



```
  self.addEventListener("message", function (e) {
          function callback() {
            returnResult({ args: [].slice.call(arguments) });
          }
          function returnResult(e, r) {
            postMessage({ cmd: "result", token: data.token, result: e
}, (hasTransferSupport && r) || []);
          }
          function extractTransfers(e) {
            var r = e[e.length - 1];
            if ("[object Array]" !== toString.call(r)) throw
Error("Operative: callback.transfer() must be passed an Array of
transfers as its last arguments");
            return r;
          }
          var data = e.data;
          if ("string" == typeof data && 0 === data.indexOf("EVAL|"))
return eval(data.substring(5)), void 0;
```

74. Poojaaachdjfkiggckefbngdegikcdob TP
from cs_window_eventListener_message to eval_sink

```
window.addEventListener("message", async (message) => {
  if (message.origin != location.origin) return;
  if (typeof message.data == "object") {
    if (!message.data.from_ch4ng34bl3) return;
    if (message.data.action == "eval") eval(message.data.content);
```