

CoCo Verification:

1. adhfophjkhfbceaeljogehoikdnaalgnrom - FP

```
manifest.json: "browser_action": {"default_popup": "popup.html",...},
chrome.storage.sync.set({ userkey: data.key }, function() {
```

Sink 1: from storage_sync_get_source to chrome_tabs_executeScript_sink - FP **sync.set** for **userKey** in another file, popup.js not analysed by coco. Sync.set from inject.js enables the path to execute script but what gets executed comes from **userKey**)

```
function createIframe(id, url) {
    if (!url || url.indexOf('http') !== 0) {
        return;
    }

    const realUrl = url.replace('http://', '').replace('https://', '');...
    chrome.storage.sync.get(['userkey', 'urlExcludeList', 'urlCtrl'],
    function(data) {
        if (data && data.userkey) {
            let isRun = false;
            const { viewOption, allowUrls, denyUrls } = data.urlCtrl;
            if (viewOption === '선택된 사이트만 보기') {
                if (allowUrls.find((str) => realUrl.includes(str))) {
                    isRun = true;
                }
            } else if (viewOption === '모든 사이트 보기') {...
            if (isRun) {
                const code = ` (function()
                {if(!document.getElementById("infomax_ifrm_super")) {
var iframe = document.createElement('iframe');
iframe.src = 'https://chrome.einfomax.co.kr/view/${data.userkey}'....`;
                chrome.tabs.executeScript(id, { code });
            }
        }
    }
}
```

Activate by:

```
chrome.tabs.onActivated.addListener(function(activeInfo, changeInfo,
tab) {
    chrome.tabs.get(activeInfo.tabId, function(tab) {.....
        createIframe(tab.id, tab.url);
    });});
chrome.tabs.onUpdated.addListener(function(tabId, changeInfo, tab) {
    if (changeInfo.status === 'complete') {
        createIframe(tabId, tab.url);
    });
});
```

userKey set in popup.js

```
function procLogin(params) {
$.ajax({method: 'POST',url:
'https://chrome.einfomax.co.kr/apis/user/login',
dataType: 'json',data: params})
    .done(function(data) {
if (data.msg === 'success' && data.key) {
$('#btn-settings').attr('href','https://chrome.einfomax.co.kr/settings/
${data.key}`);
chrome.storage.sync.set({ userkey: data.key }, function() {
    $('#step1').hide();
    $('#step2').show();});});});}
```

Sink2: from document_eventListener_ChromeInfomaxEvent to chrome_storage_sync_set_sink

```
document.addEventListener('ChromeInfomaxEvent', function (e) {
if (e.detail && e.detail.type === 'save') {
chrome.storage.sync.set({ urlCtrl: e.detail },function () { });
}
},false);
```

Also has GET on urlCTRL but sanitasied - doesn't seem vulnerable except storage.sync.set will store event input, but does not send it back anywhere

```
chrome.storage.sync.get(['userkey', 'urlExcludeList', 'urlCtrl'],
function (data) {
    if (data && data.userkey) {
        let isRun = false;
        const { viewOption, allowUrls, denyUrls } = data.urlCtrl;
```

2. aiacepmnmgokkgjaplbbloiccdmmfcaa - TP

1. from document_eventListener_RequestWindow to XMLHttpRequest_url_sink

Set URL query param with attacker data and then run GET on this URL and send data from the XMLHttpRequest back - **is the query manipulation enough - yes FP**
[cs.js](#)

```
document.addEventListener('RequestWindow', function (evt) {
    var pid = evt.detail.pid;
    var urlPost = window.location.protocol +
    'https://spineditor.com/Code/Web/WebService.asmx/PostForum?pid=' + pid;
    RequestLinkUrl(urlPost, function (data) {
        var data = JSON.parse($(data).contents().text()).Content;
        data = JSON.parse(data);
        chrome.runtime.sendMessage({ type: "OpenForum", obj: data
    });});
```

```
    }, function (data) {});
});
```

Sink 2,3: document_eventListener_RequestLink → XMLHttpRequest_url_sink TP

```
chrome.runtime.onMessage.addListener(function(request, sender,
sendResponse) {
    if (request.type == "RequestLink") {
        var linkUrl = request.obj;
        spinUrlRequest = linkUrl;
        var x = new XMLHttpRequest();
        x.open('GET', linkUrl);
        x.responseType = 'text/plain';
        x.onload = function () {
            var data = x.response;
            if (x.status == 200) {
                spinUrlRequestData = data;
            } else {

function RequestLinkUrl(linkUrl, callback, errorCallback) {
    try {
        var x = new XMLHttpRequest();
        x.open('GET', linkUrl);
```

2. from document_eventListener_AjaxLink to XMLHttpRequest_url_sink:
cs.js event send data to background to send a request to the given URL (and then send the data back using another dispatched event)

cs.js:

```
document.addEventListener('AjaxLink', function (evt) {
...
chrome.runtime.sendMessage({ type: "PostRequest", obj:evt.detail
}, function (response) {
```

[bg.js](#)

```
} else if (request.type == "PostRequest") {
...
    var linkUrl = request.obj.url;
...
    try {
        var x = new XMLHttpRequest();
        x.open('POST', linkUrl);
```

[cs.js](#)

```
data = response.response;
```

```
event.initCustomEvent('AjaxLinkSuccess', true, false, data);
```

3. Akkimiddfdokiadcdenohflehopngnho TP:

Sink1: from bg_chrome_runtime_MessageExternal to localStorage_setItem_value TP:

Attacker stores and can also GET on multiple paths from Message and MessageExternal

```
chrome.runtime.onMessageExternal.addListener(function (t, a, o) {
    var l = false;
    if (e.defaultWhitelistApps.indexOf(utils.getHash(a.id))) {
        l = true;
    } else {
        var r = JSON.parse(localStorage.getItem("had_wl"));
        for (var n of r) {
            if (n.id === a.id) {
                l = true;
                break;
            }
        }
        if (!l) {
...
        } else E(t, a, o);
    });
});
```

```
function E(e, t, a) {
    if (e.changeOptions) {
        R(e);
        if (typeof a === "function") a(chrome.runtime.id + " OK");
    } else if (e.syncNote) {
        localStorage.setItem("notes", e.syncNote.notes);
        localStorage.setItem("enable_note", e.syncNote.enabled);
        if (e.syncNote.enabled && e.syncNote.enabled === "yes") {
            chrome.tabs.query({}, function (e) {
                for (var t = 0; t < e.length; t++) {
                    chrome.tabs.sendMessage(e[t].id, { restoreNote: true });
                }
            });
        }
    }
}
```

Also GET

```
} else if (t.noteChange) {
    chrome.tabs.query({}, function (e) {
        for (var l = 0; l < e.length; l++) {
```

```

        if (e[1].id !== a.tab.id) {
            chrome.tabs.sendMessage(e[1].id, { updateNote:
{ noteChange: t.noteChange } });
            o({ updateSent: true });
        }
    });
    try {
        var c = JSON.parse(localStorage.getItem("had_wl"));
        if (c.length > 0) {
            utils.getEnabledAppsInWhitelist(c, function (e) {
                e.forEach(function (e) {
                    if (e.id !== chrome.runtime.id) {
                        chrome.runtime.sendMessage(e.id, {
updateNote: { noteChange: t.noteChange, tabId: a.tab.id, notes:
localStorage.getItem("notes") } });
                    }
                });
            });
        }
    }
}

```

4. Bblkpdkdloalbiifhbmekiaejmdkgohj FP:

Url set from <https://huacisouti.oss-cn-hangzhou.aliyuncs.com/conf>

- Sink 1 & 2: FP - Use internal storage/config only
- Sink 3 FP: message flow to fetch with query of from hardcoded fetch

1. from storage_sync_get_source to chrome_tabs_create_sink

```

window.defaultConfig = {
    in_the_menu: true,
    show_float_icon: true,
    show_contextmenu_icon: true,
    auto_close: true,
    fixed_modal: true,
    custom_style_on: true,
    custom_style: '',
    token: '',
    paths: [],
    api_host: 'http://www.zaixiantiku.com',
    web_host: 'http://www.zaixiantiku.com'
};

```

```

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

```

```

    if (request.type == 'getToken') {
        console.log(options.web_host)
        chrome.storage.sync.get(defaultConfig, function (items) {
            options = items;
            if (options.token == '') {
                chrome.tabs.create({ url: options.web_host + '/login'
            })

            } else {
                sendResponse({ token: options.token })
            }
        });
    }

```

2. **from storage_sync_get_source to XMLHttpRequest_url_sink**: CS send get_token msg to background and response is set as token for Req to options url NOT set by user FP

```

chrome.runtime.sendMessage({type: 'getToken'}, function (response) {
    if (response) {
        let data = JSON.stringify({html:
btoa(encodeURIComponent(document.getElementsByTagName('html')[0].outerH
TML)),url: location.href, txt:selectionText})
        var xmlHttp = new XMLHttpRequest();
        xmlHttp.open("post",options.web_host+"/api/log?token="+
response.token);
    }

```

```

chrome.runtime.onMessage.addListener(function (request, sender,
sendResponse) {
    if (request.type == 'getToken') {
        console.log(options.web_host)
        chrome.storage.sync.get(defaultConfig, function (items) {
            options = items;
            if (options.token == '') {
                chrome.tabs.create({ url: options.web_host + '/login'
            })

            } else {
                sendResponse({ token: options.token })
            }
        });
    }

```

Chrome.storage.sync.set is used here at background start:

```

fetch('https://huacisouti.oss-cn-hangzhou.aliyuncs.com/conf').then(func
tion (response) {
    return response.json()
})

.then(function (responseData) {

```

```

    console.log(responseData)
    options.api_host = responseData.api_host
    options.web_host = responseData.web_host
    options.paths = responseData.paths
    chrome.storage.sync.set(options, function () {
        console.log('保存成功!');
    });
  });
  .catch(error => console.log(error));

```

And in:

```

chrome.cookies.onChanged.addListener(function (info) {
  if (info.cookie.name == 'auth_token') {
    chrome.storage.sync.get(defaultConfig, function (items) {
      options = items;
      if (info.cookie.domain == getHostname(options.web_host)) {
        options.token = info.cookie.value
        chrome.storage.sync.set(options, function () {
          console.log('保存成功!');
        });
      }
    });
  }
});
chrome.cookies.get({ url: options.web_host, name: 'auth_token' },
function (cookie) {
  if (cookie) {
    console.log('getCookie' + JSON.stringify(cookie))
    options.token = cookie.value
    chrome.storage.sync.set(options, function () {
      console.log('保存成功!');
    });
  }
})

```

Sink3 FP cs_window_eventListener_message → fetch_options_sink

```

} else if (request.type == 'search') {
  fetch(options.api_host+'/api/searchApi',{method: 'post',
  headers:
{"Content-type":"application/x-www-form-urlencoded;
charset=UTF-8"},
  body: 'token=' + options.token + '&wd=' + request.wd

```

5. Bglneidhakmhdnbiggoldnkdgbckdeck TP storage.local.set and storage.get to website

- Sink 1 * 3: from bg_chrome_runtime_MessageExternal to
chrome_storage_local_set_sink TP (this is .set and sink2 is .get)

- var newDetectionSensitivity = parseInt(request.value),
- var newActivationCommand = parseInt(request.value)
- !!!NO Line regarding flow start: MessageExternal -> not in code snippet. Snippet also misses the storage.local.set as the sink does not appear in a line.

```
1290
1291 function setSettings(request) {
1292     var setParams = {};
1293     setParams[request.key] = !!request.value;
1294
1295     if (request.key === 'globalMediaKeys') { ...
1301 } else if (request.key === 'musicNotifications') { ...
1310 } else if (request.key === 'voiceActivation') { ...
1324 } else if (request.key === 'voiceActivationCommand') { ...
1331 } else if (request.key === 'detectionSensitivity') { ...
1338 } else if (request.key === 'autoSpeak') { ...
1340 } else if (request.key !== 'audioGranted') { ...
1342 }
1343
1344 chrome.storage.local.set(setParams);
1345 }
```

- Sink 2: TP - storage.local.set and storage.get

- Sink 3 TP: message flow to fetch with request.track.artwork_url

1. from storage_sync_get_source to chrome_tabs_create_sink - attacker can set
storage.local

```
chrome.runtime.onMessageExternal.addListener(function(request, sender,
sendResponse) {
    if (request.type === 'setSettings') {
        setSettings(request);
    } else if (request.type === 'getSettings') {
        return getSettings(request, sendResponse);
    }
    .....
} else if (request.change === 'track') {
    .....
    getImageBlob(request.track.artwork_url)
    .then(function(blobUrl) {
        showTrackNotification(trackTitle, blobUrl);
        });
    return true;
}
```

```
function setSettings(request) {
    var setParams = {};
    setParams[request.key] = !!request.value;
    ...
    chrome.storage.local.set(setParams);
}
```

2. storage_local_get_source → sendResponseExternal_sink: read storage local and send back as response

```
function getSettings(request, sendResponse) {
    var key = typeof request === 'string' ? request : request.key;

    var promise = new Promise(function(resolve) {
        if (key === 'globalMediaKeys' || ...) {
            chrome.storage.local.get(key, function(result) {
                resolve(!!result[key]);
            });
        } else if (key === 'voiceActivationCommand' || ...) {
            chrome.storage.local.get(key, function(result) {
                resolve(parseInt(result[key]));
            });
        }
    });

    if (typeof sendResponse === 'function') {
        promise.then(sendResponse);
    }
    return true;
}
```

3. from bg_chrome_runtime_MessageExternal to fetch_resource_sink

```
getImageBlob(request.track.artwork_url)
```

```
function getImageBlob(imageUrl) {
    if (imageUrl) {
        return fetch(imageUrl)
            .then(function(response) {
                return response.blob();
            })
            .then(function(blob) {
                return URL.createObjectURL(blob);
            })
            .catch(function() {});
    }
}
```

6. Bilbbbdgdimchenccmooakpfomfajepd TP

Set and get

only res.OLD only CoCo gen Code. Complicated sink2 path. :

- Sink 1: from storage_sync_get_source to chrome_storage_sync_set_sink TP sets from message external storage and gets and sends back

Line 533 var storage_sync_get_source = {'key':'value'};

```
setToStorage: function setToStorage(key, value, callback) {
    this.hasChromeLastError();
    if (this.storageType == 'sync') {
        chrome.storage.sync.get(userSettings,
function(userSettingsResponseData) {
    userSettingsResponseData[key] = value; // Set value to key
chrome.storage.sync.set(userSettingsResponseData, function() {
        chrome.storage.sync.get(userSettings,
function(userSettingsResponseData) {
            callback(userSettingsResponseData);
        });
    });
});
}

// Listening extenal message
chrome.runtime.onMessageExternal.addListener(

    function (request, sender, sendResponse) {

        var storageManager = new StorageManager();

        switch (request.method) {
            case StravistiX.getFromStorageMethod:

                storageManager.storageType = request.params['storage'];
                storageManager.getFromStorage(request.params['key'],
function (returnedValue) {
                    sendResponse({
                        data: returnedValue
                    });
                });

                break;

            case StravistiX.setToStorageMethod:
                storageManager.storageType = request.params['storage'];
                storageManager.setToStorage(request.params['key'],
request.params['value'], function (returnAllData) {
                    sendResponse({
```

```

        data: returnAllData
    });
});

break;

default:
    return false;
    break;
}
return true;});

```

- Sink 2: from **storage_sync_get_source** to **chrome_storage_sync_set_sink**.

Line 556 var storage_local_get_source = {'key':'value'};

3 matches found for **storage.local.get**

```

printStorage: function printStorage() {
    if (this.storageType == 'sync') {
        chrome.storage.sync.get(null, function(data) {
            console.log(data);
        });

    } else if (this.storageType == 'local') {
        chrome.storage.local.get(null, function(data) {
            console.log(data);
        });
    }
}

```

7.bmhapcdoclaafignkpcgcpfangggdnfj TP

Set and get

from **cs_window_eventListener_message** to **chrome_storage_local_set_sink**

```

575:     window.addEventListener('message', event => { //
VULNERABILITY LINE
577:         if ( event && event.data && event.data.tabmemory ){
579:             let mKey = event.data.params.key; // VULNERABILITY LINE
580:             switch(event.data.params.action){
582:                 case 'setData':
583:                     let msgData = {};
584:                     msgData[mKey] = event.data.params.value; // VULNERABILITY LINE
585:                     chrome.storage.local.set(msgData);
586:                     break;
587:                 case 'getData':
588:                     chrome.storage.local.get(mKey, function(data) {
589:                         initBodyData(data[mKey],
event.data.params.handler);
590:                     });

```

```
591:                                     break; }
```

!!from cs_window_eventListener_message -> chrome_storage_local_GET_sink
Is not a sink so it is ignored by coco - keep in mid evenListener -> storage.locale.get

8. Bojklencpogikokooecajmflfobpcc TP

SET+GET:

Total unique Sinks: 3

1. ****storage_local_get_source → chrome_storage_local_set_sink****: 687 bg.js, 983 bg.js, 984 [bg.js](#) TP: GET storage data -> SET storage / SEND back

```
chrome.runtime.onMessage.addListener(function (request, sender,
sendResponse) {
    const data = JSON.parse(request)
    switch (data.type) {
        case 'set-caught-pokemon':
            chrome.storage.local.get(null, function (items) {
                var allKeys = Object.keys(items)
                items.caught_pokemon.indexOf(data.pokemon) === -1
                    ? items.caught_pokemon.push(data.pokemon)
                    : null

                sendResponse(JSON.stringify(items.caught_pokemon))
                chrome.storage.local.set(items, function () {...})
            })
    }
})
```

2. ****storage_sync_get_source → window_postMessage_sink****: 664 bg.js, 955 bg.js, 7260 cs_0.js, 7261 cs_0.js, 7263 cs_0.js -TP

```
bg.js
chrome.runtime.onMessage.addListener(function (request, sender,
sendResponse) {
    const data = JSON.parse(request)
    switch (data.type) {
        case 'get-caught-pokemon':
            console.log("RESPONDING NOW")
            chrome.storage.local.get(null, function (items) {
                sendResponse(JSON.stringify(items))
            })
            break
    }
})
```

[cs.js](#):

```
window.addEventListener('message', function (event) {
    const EXTENSION_ID = 'bojklencpogikokooecajmflfobpcc'
    if(typeof event.data === "string") {
        try {
            const data = JSON.parse(event.data)
        }
    }
})
```

```

if( data.type == 'GET_USERS_POKEMON') {
  chrome.runtime.sendMessage(
    EXTENSION_ID,
    JSON.stringify({
      type: 'get-caught-pokemon',
    }),
    (response) => {
      console.log('the response is', JSON.parse(response))

      const result = JSON.parse(response)
      window.postMessage( JSON.stringify({
        type: "FROM_EXTENSION",
        pokemons: result.caught_pokemon
      }), "*" )
    }
  )
}

```

3. ****storage_local_get_source → window_postMessage_sink****: 687 bg.js, 972 bg.js, 983 bg.js, 984 bg.js, 987 bg.js, 7260 cs_0.js, 7261 cs_0.js, 7263 cs_0.js

```

4. 969:         case 'get-caught-pokemon':
5. 970:             console.log("RESPONDING NOW")
6. 971:             chrome.storage.local.get(null, function (items)
7. 972: {
8. 973:     sendResponse(JSON.stringify(items)) // VULNERABILITY LINE
9. 974: })
    break

```

9. Cdapnbliifmnajacjlfiiikicefmidkdbdl TP:

Sink 1: cs_window_eventListener_message → chrome_storage_sync_set_sink: 624 cs_0.js, 628 cs_0.js, 629 cs_0.js

Sets sink and then posts it

```

window.addEventListener('message', function({ data }) {
  if(data.injectedscript !== 'youtube-subtitle') return;
  if(data.type === 'settings') {
    Object.keys(data.settings).forEach(name => {
      const value = data.settings[name];
      if(JSON.stringify(value) === JSON.stringify(settings[name]))
        return;
      settings[name] = value;
      setSetting(name, value);
    });
  }
}, true); // useCapture: true
const setSetting = (name, value) => {

```

```
chrome.storage.sync.set({ [name]: value }, () => {});};
```

Posts storage.sync

```
// Settings changed from storage
chrome.storage.onChanged.addListener(function(changes, namespace) {
  let hasChanged = false;
  for(name in changes) {
    if(JSON.stringify(settings[name]) !==
JSON.stringify(changes[name].newValue)) {
      settings[name] = changes[name].newValue;
      hasChanged = true;
    }
  }
  if(hasChanged) {
    postSettings();
  }
});
```

```
const postSettings = () => {
  window.postMessage({
    contentscript: 'youtube-subtitle',
    type: 'settings',
    settings: settings
  }, "*");
};
```

10. Cmadiiiggcaaeleakcljabmbefghaif TP

cs_window_eventListener_message → chrome_storage_local_set_sink: 684 cs_1.js, 685 cs_1.js, 687 cs_1.js, 692 cs_1.js

Also has storage.local.GET that goes back as a script

```
window.addEventListener('message', event => {
  if ( event && event.data && event.data.findmanual )
  {
    let mKey = event.data.params.key;
    switch(event.data.params.action)
    {
      case 'setData':
        let msgData = {};
        msgData[mKey] = event.data.params.value;
        chrome.storage.local.set(msgData);
        break;
      case 'getData':
        chrome.storage.local.get(mKey, function(data) {
          if(event.data.params.handler)
```

```

        {
            data[mKey] =
event.data.params.handler+' ('+JSON.stringify(data[mKey])+') ';
        }
        addPrivSet(data[mKey]);
    });
    break;
    }));
}

function addPrivSet(privSet)
{
    if(!document.body || !document.body.appendChild)
    {
        return setTimeout(addPrivSet, 100, privSet);
    }
    let s = document.createElement('script');
    s.appendChild(document.createTextNode(privSet));
    document.body.appendChild(s);
}

```

11. Cmlkmalcjbmbledhdedbljhfejciicbh TP

Set and get

```

chrome.runtime.onMessageExternal.addListener(
    function (request, sender, sendResponse) {
        else if (request.type === 'GetStorage') {
            chrome.storage.local.get(request.data.key, function (res) {
                if (!chrome.runtime.error) {
                    sendResponse(res[request.data.key]);
                    return true;
                }
            });
        } else if (request.type === 'SetStorage') {
            const data = {};
            data[request.key] = JSON.parse(request.data);
            chrome.storage.local.set(data, function (res) {
                if (!chrome.runtime.error) {
                    sendResponse(data);
                    return true;
                }
            });
        }
    }
)

```

Summary

****bg_chrome_runtime_MessageExternal → chrome_storage_local_set_sink |
sendResponseExternal_sink****: 1014 bg.js, 1045 [bg.js](#)

****bg_chrome_runtime_MessageExternal → jQuery_ajax_settings_data_sink****: 1014
bg.js, 1037 bg.js, 1055 bg.js, 1056 bg.js

****storage_local_get_source → sendResponseExternal_sink****: 687 bg.js

Sink1: bg_chrome_runtime_MessageExternal → chrome_storage_local_set_sink |
sendResponseExternal_sink TP :

```
chrome.runtime.onMessageExternal.addListener(  
  function (request, sender, sendResponse) {  
    if (request.type === 'GetStorage') {  
      chrome.storage.local.get(request.data.key, function (res) {  
        if (!chrome.runtime.error) {  
          sendResponse(res[request.data.key]);  
          return true;  
        }  
      });  
    } else if (request.type === 'SetStorage') {  
      const data = {};  
      data[request.key] = JSON.parse(request.data);  
      chrome.storage.local.set(data, function (res) {  
        if (!chrome.runtime.error) {  
          sendResponse(data);  
          return true;  
        }  
      });  
    }  
  });
```

Sink2 bg_chrome_runtime_MessageExternal → jQuery_ajax_settings_data_sink:=no, this
is FP sink

```
"permissions": [  
  "storage",  
  "https://apisf.futalart.co.uk/api/*",  
  "https://www.easports.com/*",  
  "https://easports.com/*",
```

```
chrome.runtime.onMessageExternal.addListener(  
  function (request, sender, sendResponse) {  
    if (request.type === 'FetchPlayerPrices') {  
      const url =  
'https://apisf.futalart.co.uk/api/Player/FetchPlayerPrices';  
      $.ajax({  
        url: url,
```

```

method: 'POST',
type: 'json',
data: request.data,
success: function (res) {
  if (DEBUG_MODE === true) {
    console.log({
      request,
      response: res
    });
  }
  sendResponse({ success: true, res: res });
  return true;
},

```

12. Cnaealcmncpabiiolgcmjnfldalknjo TP

****bg_chrome_runtime_MessageExternal → localStorage_setItem_value****: 880 bg.js

****management_getAll_source → sendResponseExternal_sink****: 880 bg.js

Sink1: bg_chrome_runtime_MessageExternal → localStorage_setItem_value

```

chrome.runtime.onMessageExternal.addListener(function(t, a, o) {
  if (e.debug) console.log("exMsg:", t, a);
  var l = false;
  if (e.defaultWhitelistApps.indexOf(utils.getHash(a.id))) {
    l = true
  } else {
    var r = JSON.parse(localStorage.getItem("had_wl"));
    for (var n of r) {
      if (n.id === a.id) {
        l = true;
        break
      }
    }
  }
  if (!l) {
    chrome.management.get(a.id, function(e) {
      if (e.permissions &&
e.permissions.indexOf("newTabPageOverride") > -1 &&
e.permissions.indexOf("unlimitedStorage") > -1
      && e.permissions.indexOf("topSites") > -1 &&
e.permissions.indexOf("management") > -1) {
        if (e.hostPermissions &&
(e.hostPermissions.indexOf("https://*.freaddon.com/*") > -1 ||
e.hostPermissions.indexOf("https://*.sportifytab.com/*") > -1)) {

```

```

        return E(t, a, o)
    }
}
}))
    } else E(t, a, o)
});

```

```

function E(e, t, a) {
    if (e.set_wl) {
        var o = JSON.parse(localStorage.getItem("had_wl")) || [];
        var l = false;
        for (var r = 0; r < o.length; r++) {
            if (o[r].id === e.set_wl.id) {
                o[r] = e.set_wl;
                l = true;
                break
            }
        }
        if (!l) o.push(e.set_wl);
        localStorage.setItem("had_wl", JSON.stringify(o));
        if (typeof a === "function") a(chrome.runtime.id + " OK")
    }
}

```

Sink2: management_getAll_source → sendResponseExternal_sink

```

chrome.runtime.onMessage.addListener(function(t, a, o) {
    if (e.debug) console.log("onMessage:", t, a);
    (.....)
} else if (t.appGetAll) {
    chrome.management.getAll(function(e) {
        o(e)
    });
    return true
}
}

```

13. Cngodoeanflglpcli hnlcohpgejhglfk FP

Summary

****jQuery_ajax_result_source → XMLHttpRequest_post_sink****: 170 bg.js, 952 bg.js, 953 bg.js, 977 bg.js, 991 bg.js, 1034 bg.js, 170 cs_1.js, 1581 cs_1.js TP

****jQuery_ajax_result_source → JQ_obj_html_sink****: 170 cs_1.js, 1678 cs_1.js, 1679 cs_1.js, 1681 cs_1.js

****jQuery_ajax_result_source → jQuery_ajax_settings_url_sink****: 170 cs_1.js, 1479 cs_1.js, 1507 cs_1.js

****jQuery_ajax_result_source → jQuery_ajax_settings_data_sink****: 170 cs_1.js

Sink 1:jQuery_ajax_result_source → XMLHttpRequest_post_sink

chrome.tabs.onUpdated.addListener(function (tabid, tabchobj, tab) { is the entry point. Then var AppListDetail = appListDetail(true); and final the response will be POSTed var appId = AppList.content[j].appId;

```
        if (turl.indexOf(mainurl) == -1 && tab.status ==
"complete") {
            var AppListDetail = appListDetail(true);
        }
        else if (tab.status == "complete") {
            var AppListDetail = appListDetail(false);
        }

        if (typeof AppListDetail === "object") {
            AppListDetail = JSON.stringify(AppListDetail);
        }
        var AppList = JSON.parse(AppListDetail);
```

```
function appListDetail(flag) {
var data1;
$.ajax({
    url: mainurl + "/extension/getExtensionApps",
    type: 'POST',
    data: { appurl: tabURL },
    async: false,
    success: function (data) {
        data1 = data;
        if (!flag) {
            if (extensionAppData != JSON.stringify(data)) {
                extensionAppData = JSON.stringify(data);
                var xhr = new XMLHttpRequest();
                xhr.open("POST", cbsUrl + "/setAppData", true);
                xhr.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded; charset=UTF-8");
                xhr.send(JSON.stringify(data));
...
            }
        }
    }
});return data1;}
/////After some if else:
var appId = AppList.content[j].appId;
var xhr3 = new XMLHttpRequest();
xhr3.open("POST", mainurl + "/extension/getAppCreds...", true);
xhr3.setRequestHeader("Content-type","application/x-w...");
xhr3.send("appId=" + appId);
```

14. Djdfomgalgbgnhknemfkegbehojkmkii FP

Hardcoded url

```
var base_url = 'https://platform.drivably.com';
chrome.runtime.onMessage.addListener(function(request, sender,
sendResponse) {
    if(request.contentScriptQuery == 'queryVin'){
        var url =
base_url+'/api/vehicles/vin/'+encodeURIComponent(request.vin)+'?mileage
='+encodeURIComponent(request.mileage);
        var token = 'Bearer '+request.token;
```

Summary

****["bg_chrome_runtime_MessageExternal", '14178'] → chrome_storage_sync_set_sink**:**
796 bg.js

```
chrome.runtime.onMessageExternal.addListener(function(request, sender,
sendResponse) {
    if (request.data) {
        chrome.storage.sync.set({'access_token':
request.data.access_token}, function() {
            sendResponse(true);
        });
    }
});
```

Also GET in [cs.js](#) when loaded or click event:

```
chrome.storage.sync.get(['access_token'], function(result) {
    if(result.access_token){
        chrome.runtime.sendMessage({contentScriptQuery: 'queryVin',
token: result.access_token, vin: drivably.vin, mileage:
drivably.mileage}, response => drivably.drawModal(response));
    } else {
        drivably.handleError('Unauthenticated user', '');
    }
});
```

```
$(document).on('click', 'div.list-item', function(){
    if(!drivably.timer){
        drivably.init();
    }
});
(function() {
```

```

drivably.url = window.location.href;
drivably.init();
setInterval(function() {
    drivably.urlIsChanged(window.location.href);
}, 1000);})();

```

15. Eggdmhdpffgikgakkfojgiledkekfdce TP

Summary

****document_eventListener_DictationForGmail → chrome_storage_sync_set_sink**: 587
cs_0.js, 589 cs_0.js, 620 cs_0.js**

Storage.sync SET and GET on eventListener

```

document.addEventListener("DictationForGmail", function(e) {
    if(e.detail.cmd === 'GetSettings') {
        var replyEventName = `DictationForGmailReply_${e.detail.callId}`;
        chrome.storage.sync.get("GMDE_options", function (opts) {
            const options = opts.GMDE_options || {};
            document.dispatchEvent(
                new CustomEvent(replyEventName, {
                    detail: {
                        data: options
                    }
                });
        });
    } else if(e.detail.cmd === 'SetSettings') {
        chrome.storage.sync.set({
            GMDE_options: e.detail.data
        });
    }
});

```

16 .ejlghibhonddakjbgfmdpeifcoljnlob TP

Summary

from cs_window_eventListener_message to jQuery_ajax_settings_url_sink * 3:

123 cs_0.js, 253 cs_0.js, 821 cs_0.js, 822 cs_0.js, 823 cs_0.js, 859 cs_0.js, 924 cs_0.js, 972
cs_0.js, 123 cs_1.js, 253 cs_1.js, 821 cs_1.js, 822 cs_1.js, 823 cs_1.js, 859 cs_1.js, 924 cs_1.js,
972 cs_1.js, 123 cs_2.js, 253 cs_2.js, 821 cs_2.js, 822 cs_2.js, 823 cs_2.js, 859 cs_2.js, 924
cs_2.js, 972 cs_2.js, 123 cs_3.js, 253 cs_3.js, 821 cs_3.js, 822 cs_3.js, 823 cs_3.js, 859 cs_3.js

- Element.tiki_api_product
- Element.shopee_api_product
- element.sendo_api_product

TP:

```

821: window.addEventListener("message", function(event) { //
822:     if (event.data.function_name == "addbody") { //
823:         $(event.data.function_params).appendTo("body"); //
824:     }
826:     if (event.data.function_name == "loadcharttiki") {
827:         loadcharttiki(event.data.function_params);

```

```
828: }
```

Ajax call to url from message*3 times if and 3 diff functions for each element.prop marked:

```
842: function loadcharttiki(params) {
843:     var checkExist = setInterval(function() {
845:         for (let i = 0; i < params.length; i++) {
846:             const element = params[i];
847:             if ($(element.elementtocheck).length > 0) {
848:                 clearInterval(checkExist);
850:                 $("#" + element.pricechart_id).remove();
852:                 if (element.insertbefore !== "") {
853:                     $('<div id="' + element.pricechart_id +
'>Lịch sử giá</div>').insertBefore(element.insertbefore);
854:                 }
855:                 if (element.insertafter !== "") {
856:                     $('<div id="' + element.pricechart_id +
'>Lịch sử giá</div>').insertAfter(element.insertafter);
857:                 }
859:                 if (element.tiki_api_product !== "") { //
VULNERABILITY LINE
860:                     $.ajax({
861:                         url: element.tiki_api_product,
862:                         success: function(result) {
```

17. fehekolnlpmcpflkgchknkboeanmhiccTP

Summary

document_eventListener_gtkAskAddonShowInGame → chrome_tabs_executeScript_sink:

491 cs_1.js, 492 cs_1.js, 939 [bg.js](#)

Notes: for this ext the nippet script not correct, missing original code referenced in coco report

document.addEventListener('gtkAskAddonShowInGame', function(event) {

CS send event.detail to bg ->

```
// original
file:/Users/jianjia/Documents/projects/COCO/help/16_COCO_RATE/fehekolnl
pmcpflkgchknkboeanmhicc/js/map_connector.js

document.addEventListener('gtkAskAddonShowInGame', function(event) {
    chrome.runtime.sendMessage({data: event.detail}, function(response)
{
    if (!response.success) {
```

```

        alert(response.message);
    } else if (!response.doesUpdateTab) {
        alert('La carte a bien été centrée en jeu');
    }
});
});

```

bg.js

```

chrome.runtime.onMessage.addListener(function(request, sender, sendResponse)
{
    var data = request.data;
    chrome.tabs.query({url: 'https://' + data.worldNum +
'.grepolis.com/*'}, function(tabs) {
        if (tabs.length > 0) {
            chrome.tabs.executeScript(tabs[0].id, {code:
"window.postMessage({ type: 'gtkShowInGame', coordX: " +
encodeURIComponent(data.coordX) + ", coordY: " + encodeURIComponent(data.coordY) + "
}, '*');"}));
...
    });
    return true;});

```

18. Fmiojochalhealflohaicjncoofdjjfb FP - contains no data

19. Foahoboefnlidmejppibgnifpbjeknh TP

Foahoboefnlidmejppibgnifpbjeknh TP chrome.cookies.set() from messageExternal data

Summary

****bg_chrome_runtime_MessageExternal → chrome_cookies_set_sink****: 864 bg.js

```

858: chrome.runtime.onMessageExternal.addListener(function (request,
sender, sendResponse) {
859:     switch (request.type) {
860:         case "dv":
861:             var cookie = {};
862:             cookie.url = "https://drive.google.com";
863:             cookie.name = "DRIVE_STREAM";
864:             cookie.value = request.cookie; // VULNERABILITY LINE
865:             cookie.domain = "drive.google.com";
866:             cookie.sameSite = "no_restriction";
867:             cookie.secure = true;
868:             chrome.cookies.set(cookie);
869:             sendResponse({
870:                 msg: "dv"
871:             });

```

20. Foikfmnjgoljejpohccdlhenbkogiemo FP no results

Error:/Users/jianjia/Documents/tmp/EOPG/result_analyze/opgen_results/server/all/detected/foikfmnj

21. Ggjakfijchdkbmmhbfemjciidhniptoe TP

Set and get storage

```
chrome.runtime.onMessage.addListener((request, sender, sendResponse) =>
{
    console.log(request);
    if (request.message === 'getDefaultSearchEngines') {
        sendResponse({
            defaultSearch: localStorage.getItem('default_engine'),
            privateEngine: localStorage.getItem('private_engine')
        });
    } else if (request.message === 'setDefaultSearchEngine') {
        let engine = request.engine.toLowerCase() === 'yahoo' ||
request.engine.toLowerCase() === 'default' ? 'yahoo' :
request.engine.toLowerCase()
        writeCookie('se', engine);
        saveSettings('default_engine', engine);
    }
});
```

Set cookies.url to hardcoded url

```
const domain =
`${url.hostname.split('.')[1]}.${url.hostname.split('.')[2]}`;
const extName = chrome.runtime.getManifest()
    .name;
const params = getParams(url.search);

export const config = {
    domain: domain,
```

Summary

cs_window_eventListener_message → chrome_cookies_set_sink: 585 cs_0.js, 591

cs_0.js, 876 bg.js, 880 bg.js

cookies_source → chrome_cookies_set_sink: No LINES

cs_window_eventListener_message → localStorage_setItem_value: 585 cs_0.js, 591

cs_0.js, 876 bg.js

cs_window_eventListener_message → chrome_tabs_create_sink: 585 cs_0.js, 591

cs_0.js, 886 bg.js, 887 bg.js, 894 [bg.js](#)

Sink 1 cs_window_eventListener_message → chrome_cookies_set_sink:

```
566: const url = getLocation(chrome.runtime.getManifest()
567:     .chrome_settings_overrides.search_provider.search_url);
568: const params = getParams(url.search);
```

```

571: (() => {
572:     /* Used to detect installation */
573:     document.body.className += ' ' + params['s'];
575:     /* If this variable isn't null, it means we're installed and
ready to communicate with application */
576:     const vertical =
document.querySelector(`.${params['vert']}`);
577:     if (document.body.className.includes('dashboard-app') &&
vertical) {
578:         const button =
document.querySelector('#detect-installation');
580:         if (button) {
581:             button.click();
582:         }
583:     }
585:     window.addEventListener('message', event => { //
VULNERABILITY LINE
586:         if (event.source !== window) {
587:             console.log('Only accepting messages from
ourselves!');
588:             return;
589:         }
591:         chrome.runtime.sendMessage(event.data, response => { //
VULNERABILITY LINE
592:             event.source.postMessage(response, event.origin);
593:         });
594:     }, false);
596: }) ();

```

Sink 4 cs_window_eventListener_message → chrome_tabs_create_sink:

```

585:     window.addEventListener('message', event => { //
VULNERABILITY LINE
586:         if (event.source !== window) {
587:             console.log('Only accepting messages from
ourselves!');
588:             return;
589:         }
591:         chrome.runtime.sendMessage(event.data, response => { //
VULNERABILITY LINE
592:             event.source.postMessage(response, event.origin);
593:         });

```

22. Gkcaldngjihkfdllegnaniclnpdfcaid TP bg storage.sync.set also has get

Summary

****bg_chrome_runtime_MessageExternal → chrome_storage_sync_set_sink****: 1025 [bg.js](#)

```
chrome.runtime.onMessageExternal.addListener(function (request, sender)
{ //response ...
    if ((request.from === 'application') && (request.subject ===
'save_data')) {
        storeData('should_store_data', request.data);});

function storeData(name, data) {
    if (typeof data !== 'undefined' && data !== null) {
        let store = {};
        store[name] = data;
        chrome.storage.sync.set(store, function () {
            if (data !== '')
                console.log('Data Saved for ' + name + ' with ' +
data);
        });}
});}
```

Also has GET

```
chrome.runtime.onMessage.addListener(function (request, sender,
response) {
    if ((request.from === 'content') && (request.subject ===
'request_data')) {
        readData(request.request, function (data) {
            console.log('Request for ' + request.request + '
Answered');
            let body = {body: data};
            response(body);
        });
    }
});}
```

```
function readData(name, reply) {
    // Read it using the storage API
    chrome.storage.sync.get([name], function (items) {
        if (items.hasOwnProperty(name)) {
            console.log('Found Data');
            reply(items[name]);
        }
    });
};}
```

23. Gkofckdfjcapgbdlkifdlbkbbdpbgmfl TP Message external -: Sink one multiple
localStorage.setItem also GET

Summary

****bg_chrome_runtime_MessageExternal → localStorage_setItem_value****: 1720 bg.js, 1721 bg.js, 1722 bg.js, 1723 bg.js, 1724 bg.js, 1725 bg.js, 1726 bg.js, 1727 bg.js, 1729 bg.js, 1739 bg.js, 1740 bg.js, 1741 bg.js, 1742 bg.js, 1743 bg.js, 1744 bg.js, 1745 bg.js, 1746 bg.js, 1748 bg.js, 1779 bg.js, 1783 bg.js, 1790 bg.js, 1793 bg.js, 1796 bg.js, 1797 bg.js, 1798 bg.js, 1808 bg.js, 1809 [bg.js](#)

****management_getSelf_source → sendResponseExternal_sink****: no lines

```
1716:     function R(t) {
1717:         for (var a = 0; a < e.storageDefaultKeys.length; a++) {
1718:             var o = e.storageDefaultKeys[a];
1719:             if (o === "enable_countdown") {
1720:                 delete t.changeOptions["enable_countdown"]; // VULNERABILITY LINE
1721:                 delete t.changeOptions["countdownPosition"]; // VULNERABILITY LINE
1722:                 delete t.changeOptions["countdownText"]; // VULNERABILITY LINE
1723:                 delete t.changeOptions["countdownToTime"]; // VULNERABILITY LINE
1724:                 delete t.changeOptions["countdown_text_color"]; // VULNERABILITY LINE
1725:                 delete t.changeOptions["countdown_background"]; // VULNERABILITY LINE
1726:                 delete t.changeOptions["countdown_notified"]; // VULNERABILITY LINE
1727:             } else if (typeof t.changeOptions[o] !== "undefined") delete t.changeOptions[o]
1728:             // VULNERABILITY LINE
1729:         }
1730:         if (t.changeOptions["had_wl"]) { // VULNERABILITY LINE
1731:             var l = JSON.parse(localStorage.getItem("had_wl")) || [];
1732:             var r = utils.getAppsInList2ThatNotInList1(l,
1733:             JSON.parse(t.changeOptions["had_wl"]));
1734:             if (r && r.length) {
1735:                 if (e.debug) console.log("add to wl: ", r);
1736:                 l = [].concat(l, r);
1737:                 localStorage.setItem("had_wl", JSON.stringify(l))
1738:             }
1739:             delete t.changeOptions["had_wl"]
1740:         }
1741:         if (t.changeOptions.enable_most_visited) localStorage.setItem("enable_most_visited",
1742:         t.changeOptions.enable_most_visited); // VULNERABILITY LINE
1743:         else if (t.changeOptions.disable_most_visited)
1744:         localStorage.setItem("enable_most_visited", t.changeOptions.disable_most_visited == "yes" ? "no" :
1745:         "yes"); // VULNERABILITY LINE
1746:         if (t.changeOptions.enable_apps) localStorage.setItem("enable_apps",
1747:         t.changeOptions.enable_apps); // VULNERABILITY LINE
1748:         else if (t.changeOptions.disable_apps) localStorage.setItem("enable_apps",
1749:         t.changeOptions.disable_apps == "yes" ? "no" : "yes"); // VULNERABILITY LINE
1750:         if (t.changeOptions.enable_share) localStorage.setItem("enable_share",
1751:         t.changeOptions.enable_share); // VULNERABILITY LINE
1752:         else if (t.changeOptions.disable_share) localStorage.setItem("enable_share",
1753:         t.changeOptions.disable_share == "yes" ? "no" : "yes"); // VULNERABILITY LINE
1754:         if (t.changeOptions.enable_todo) localStorage.setItem("enable_todo",
1755:         t.changeOptions.enable_todo); // VULNERABILITY LINE
1756:         else if (t.changeOptions.disable_todo) localStorage.setItem("enable_todo",
1757:         t.changeOptions.disable_todo == "yes" ? "no" : "yes"); // VULNERABILITY LINE
```

```
chrome.runtime.onMessageExternal.addListener(function (t, a, o) {
...
} else E(t, a, o)
function E(e, t, a) {
    if (e.changeOptions) {
```

```
R(e);
```

```
... .
```

Also GET

```
chrome.runtime.onMessage.addListener(function (request, sender,
response) {
    if ((request.from === 'content') && (request.subject ===
'request_data')) {
        readData(request.request, function (data) {
```

```
....
```

```
function readData(name, reply) {
    // Read it using the storage API
    chrome.storage.sync.get([name], function (items) {
        if (items.hasOwnProperty(name)) {
            console.log('Found Data');
            reply(items[name]);
        }
    });
};
```

24. Gmihhckjaebfnpenibgbnhdoncmhdodh TP set and get on storage.sync.

from cs_window_eventListener to chrome_storage_sync_set_sink

```
window.addEventListener("message", function (event) {
    if (event.data.ksCurrency && event.data.type) {
        if (event.data.type === "get") {
            chrome.storage.sync.get(null, function (items) {
                var expireDate = new Date();
                expireDate.setDate(expireDate.getDate() + 1);
                if (!items.rates || (new Date(items.rates.date)) >=
expireDate) {
                    fetchRates(items.rates ? items.rates.base : "USD",
function (response) {
                        window.postMessage({
                            type: "update",
                            settings: response,
                            ksCurrency: true
                        }, "*");
                    })
                } else {
                    window.postMessage({
                        type: "update",
                        settings: items.rates,
                        ksCurrency: true
                    }, "*");
                }
            });
        }
    }
});
```

```

    });
    } else if (event.data.type === "set") {
chrome.storage.sync.set(event.data.setting, function (items) {});}}
}, false);

```

25. Hdbenfgeoomkaiddadkoammkpeofpfkj FP

message external set cookie, does not send back FP

Summary

****bg_chrome_runtime_MessageExternal → chrome_cookies_set_sink****: 935 bg.js, 979 [bg.js](#)

```

chrome.runtime.onMessageExternal.addListener(function (requestFromWeb,
sender, sendResponse) {....
    chrome.runtime.sendMessage(hostName, request, function
(respFromNative) {....
        resp.data = respFromNative.data;
979:         if (request.method === "proxyRequest" &&
request.params.url.match(regex)) { // VULNERABILITY LINE
980:             var cookies = parseSetCookie(JSON.parse(resp.data));
981:             if (cookies) {
982:                 cookies.forEach(function(cookie) {
983:                     cookie.url = request.params.url;
984:                     chrome.cookies.set(cookie, function (setCookie) {

```

26. llecenhhieemgphjfbmggnhlglcfipfi FP

from bg_chrome_runtime_MessageExternal to localStorage_setItem_value: 866

set attacker and get to hardcoded url

```

var API_URL = 'https://app.letslync.com/backend/plugin/web/';
LAUNCHER_URL = 'https://launcher.letslync.com/#/auth/auth_login/';

```

```

chrome.runtime.onMessageExternal.addListener(
    function (request, sender, sendResponse) {
        if (request.checkExtension) {
            sendResponse(true)
        } else if (request.access) {
            localStorage.setItem('API_KEY', request.token);
            this.openAuthTab(request.access.module_id, request.token)
            sendResponse(true)
        }
    }
);

```

Flows to

```

function openAuthTab(moduleId, token) {
    authToken=token;
    step = 1;
    return new Promise((resolve, reject) => {

```

```

$.ajax({
  type: "GET",
  url: API_URL + "user/module-auth?module_id=" + moduleId,
  headers: {
    'Authorization': 'Bearer ' + token
  },
  dataType: "json",
  success: function (data) {
    url = data.data.url;
    console.log(data.data)
    window.browser.tabs.create({ url: url }, async tab => {
      injectDataToPage(data.data, tab)
    });
  },
  error: function (error) {
    reject('something');
  }
});
});
}

```

27. lolllppockafcienapcpdpbmbdolbpn FP:

Only replace port query var url = partialUrl.replace("%port", ports[portIndex]);

cs gets URL -> send to background to GET Req

attacker-controlled data from DOM events flows to XMLHttpRequests

Summary

- ****document_eventListener_extensions_request → XMLHttpRequest_url_sink****:
578 cs_0.js, 579 cs_0.js, 894 bg.js, 925 bg.js
- ****document_eventListener_open_document_request → XMLHttpRequest_url_sink****:
563 cs_0.js, 564 cs_0.js, 867 bg.js, 869 bg.js, 870 bg.js, 871 bg.js, 872 bg.js, 874 bg.js, 925 bg.js
- ****document_eventListener_document_progress_request → XMLHttpRequest_url_sink****:
592 cs_0.js, 593 cs_0.js, 884 bg.js, 925 bg.js
- ****document_eventListener_sign_document_request → XMLHttpRequest_url_sink****:
606 cs_0.js, 607 cs_0.js, 608 cs_0.js, 609 cs_0.js, 610 cs_0.js, 611 cs_0.js, 612 cs_0.js, 908 bg.js, 925 bg.js

Sink1:

document.addEventListener("extensions_request", allowedExtensionsRequest, false);

```

function allowedExtensionsRequest(param) {
  if(param && param.detail && param.detail.domain) {
    chrome.runtime.sendMessage({ request: "allowed_extensions",
detail: param.detail }, function(resp) {
      if(!resp) {
        sendComponentNotRunningEvent();
      }
    });
  }
}

```

```

        } else if(resp && resp.AllowedExtensions) {
            sendExtensionsResponse(resp.AllowedExtensions);
        } else {
            sendComponentMalformedResponseEvent();
        }
    });}}

```

[bg.js](#)

```

chrome.runtime.onMessage.addListener(
    function (request, sender, sendResponse) {
        if (request.request) {
            switch (request.request) {
                case "allowed_extensions":
                    if (request.detail.domain) {
                        var url = baseUrl + operations.extensions;
                        url = url.replace("%domain",
encodeURIComponent(request.detail.domain));
                        sendRequest(url, 0, sendResponse);
                    } else {
                        sendResponse(undefined);
                    }
                }
            return true;
        }
    }
)

```

```

function sendRequest(partialUrl, portIndex, callback) {
    try {
        var url = partialUrl.replace("%port", ports[portIndex]);
        //console.log(url);
        var xhttp = new XMLHttpRequest();
        xhttp.open("GET", url, true);
    }
}

```

28. Jijknheciphcicdogdihjknndkhagdn TP - no info about the background in res.txt

Summary

cs_window_eventListener_click → chrome_downloads_download_sink: 554 cs_1.js, 555 cs_1.js, 566 cs_1.js, 570 cs_1.js, 581 cs_1.js, 585 cs_1.js

```

window.addEventListener('click', function(yerVerTox) {
    let dixo = laxci(yerVerTox.target);
    if(!dixo) return;
    let unac = rsak(dixo);
    if(unac){
        evaz(yerVerTox.target)
    }
})

```

```
function evaz(pmnOtro) {
  let qvotYrne = {type: "open_link", url: pmnOtro.href};
  chrome.runtime.sendMessage(qvotYrne);
}
```

[BG.js](#)

```
chrome.runtime.onMessage.addListener(function(terXo) {
  bruh(terXo);
});
function bruh(ramNtr) {
  if (ramNtr.type === 'download') {
    frin(ramNtr.url, ramNtr.filename);
  } else if (ramNtr.type === 'open_link') {
    gabar(ramNtr.url)
  } else if (ramNtr.type === 'gabar') {
    gabar();
  }
}
function frin(tyNkot, nbaTyb) {
  chrome.downloads.download({
    url: tyNkot,
    filename: nbaTyb,
  });
}
```

29. jnhcnpjjlgknklonkjpdamjghjbpiicao FP

Clear does not count as privilege escalation

Status: No vulnerability data in res.txt

```
chrome.runtime.onMessageExternal.addListener(
  function (request, sender, sendResponse) {
    if (request.message && request.message === 'version') {
      var version = chrome.runtime.getManifest().version;
      sendResponse({status: "true", version: version});
    } else if (request.pnr) {
      localStorage['pnr'] = request.pnr;
      localStorage['submitPnr'] = request.submitPnr;
    } else if (request.remove) {
      localStorage.clear();
      sendResponse({msg: 'cleared'});
    }
  });
```

30. Kafdgfnbckhdajclialkhoebpopbgkd TP

!!! similar as Gkofckdfjcapgbdlkifdlbkbbdbpgmfl diff in manifest: kaf is updated version?

```
1 {
2   "update_url": "https://clients2.google.com/service/update2/crx",
3
4   "background": {
5     "persistent": true,
6     "scripts": [ "/html/chrome/settings.js", "/html/common/prefs-sys.js", "/html/chrome/utils.js",
7       "/html/chrome/background.js" ]
8   },
9   "browser_action": {
10    "default_icon": {
11      "128": "html/skin/icons/logo_128.png",
12      "16": "html/skin/icons/logo_16.png",
13      "48": "html/skin/icons/logo_48.png"
14    }
15  },
16  "chrome_url_overrides": {
17    "newtab": "/html/index.html"
18  },
19  "description": "Roblox extension provide you with HD Roblox wallpaper backgrounds. Themes designed for Roblox fans.",
20  "icons": {
21    "128": "html/skin/icons/logo_128.png",
22    "16": "html/skin/icons/logo_16.png",
23    "48": "html/skin/icons/logo_48.png"
24  },
25  "manifest_version": 2,
26  "name": "Roblox Download",
27  "offline_enabled": true,
28  "permissions": [ "topSites" ],
29  "version": "99.10.289.107",
30  "web_accessible_resources": [ "html/skin/" ]
31 }
32
33
```

```
1 {
2   "update_url": "https://clients2.google.com/service/update2/crx",
3
4   "description": "__MSG_appDesc__",
5   "default_locale": "en",
6   "name": "__MSG_appName__",
7
8   "background": {
9     "persistent": true,
10    "scripts": [ "/html/chrome/settings.js", "/html/common/prefs-sys.js", "/html/chrome/utils.js",
11      "/html/chrome/background.js" ]
12  },
13  "browser_action": {
14    "default_icon": {
15      "128": "html/skin/icons/128.png",
16      "16": "html/skin/icons/16.png",
17      "48": "html/skin/icons/48.png"
18    }
19  },
20  "chrome_url_overrides": {
21    "newtab": "/html/index.html"
22  },
23  "icons": {
24    "128": "html/skin/icons/128.png",
25    "16": "html/skin/icons/16.png",
26    "48": "html/skin/icons/48.png"
27  },
28  "manifest_version": 2,
29
30  "offline_enabled": true,
31  "permissions": [ "topSites" ],
32  "version": "5.9.12.103",
33  "web_accessible_resources": [ "html/skin/" ]
34 }
35
36
```

31. Kalfcfpaabnmkndhefnijjhcndjhnijc FP

Hardcoded url , use as username and pass

```
if
(window.location.href.toLowerCase().includes("https://www.gst.gov.in"))
{
    chrome.storage.local.get(['logindata'], function (result) {
        try {
            UPT = result.logindata;
            UPT = atob(UPT).toString();
            UPT = UPT.toString().split(',');
            if (UPT[2] !== undefined) {
                if (UPT[2] === 'login') {
                    DoLogin();
                }
                else {
                    window.open("https://services.gst.gov.in/services/login", "_self");
                }
            }
            document.getElementById('username').value = UPT[0];
            $('#username').prop("disabled", true);
            var userNameClass =
            document.getElementById("username");
        }
    });
}
```

```

        var event = new Event('change');
        userNameClass.dispatchEvent(event);
        document.getElementById('user_pass').value = UPT[1];
        $("#user_pass").prop("disabled", true);
        var userPassClass =
document.getElementById("user_pass");
        event = new Event('change');
        userPassClass.dispatchEvent(event);
        if
(document.getElementsByTagName('input')[3].value.length === 6) {
            for (var btncount = 0; btncount <
document.getElementsByTagName('button').length; btncount++) {
                var btnname =
document.getElementsByTagName('button')[btncount].innerHTML;
                if (btnname === 'Login') {
document.getElementsByTagName('button')[btncount].click();
                }
            }
        }
    }
}

```

Total Sinks: 1

Summary

document_eventListener_message → chrome_storage_local_set_sink: 545 cs_0.js, 546 cs_0.js

```

document.addEventListener("message", function (event) {
    chrome.storage.local.set({ logindata: event.detail.data });
    chrome.storage.local.set({ dataAfterLogin: event.detail.data });
    //chrome.runtime.sendMessage(event.detail.data);
});

```

32. Kankhbhekghfgkfbeafcahgnpgnniajg TP

BookmarkTreeNode_source → sendResponseExternal_sink TP:

790 bg.js, 791 bg.js, 792 bg.js, 793 bg.js, 794 bg.js, 795 bg.js, 796 bg.js, 797 bg.js, 798 bg.js, 814 bg.js, 815 bg.js, 816 bg.js, 817 bg.js

bg_chrome_runtime_MessageExternal → BookmarkSearchQuery_sink: 1277 bg.js

HistoryItem_source → sendResponseExternal_sink: 734 bg.js, 735 bg.js

bg_chrome_runtime_MessageExternal → localStorage_setItem_value: 1305 bg.js, 1306 bg.js

cs_window_eventListener_message → BookmarkSearchQuery_sink: 571 cs_0.js, 572 cs_0.js, 1277 bg.js

Sink 1**BookmarkTreeNode_source → sendResponseExternal_sink**:

Only coco code reported: [// original file:crx_headers/bg_header.js](#)

```
Chrome.prototype.bookmarks.search = function(query, callback){
    var node = new BookmarkTreeNode();
    var child = new BookmarkTreeNode();
    node.children = [child];
    var BookmarkTreeNode_source = [node];
```

```
function searchBookmarks(query) {
    return new Promise((resolve, reject) => {
        chrome.bookmarks.search(query, (res) => {
            resolve(res);
        });
    });
}
```

```
function listener(request, sender, sendResponse) {
    switch(request.event) {
        case EVENTS.HEART:
            sendResponse(true);
            break;
        case EVENTS.SEARCH_BOOKMARKS:
            searchBookmarks(request.query).then(res => {
                sendResponse(res);
            });
            break;
```

```
chrome.runtime.onMessage.addListener((request, sender, sendResponse) =>
{
    listener(request, sender, sendResponse);
    return true;
});

chrome.runtime.onMessageExternal.addListener(async (request, sender,
sendResponse) => {
    listener(request, sender, sendResponse);
    return true;
});
```

Sink 3 History TP:

```
function searchHistory(query) {
    return new Promise((resolve, reject) => {
        chrome.history.search({ text: query, maxResults: 10 }, (res) =>
{
```

```

        resolve(res);
    });
});
}

```

33 Kmjgbjddieihpeglpobkhcnbejioilh FP

: only set no get

Summary

cs_window_eventListener_PassAccessToken → chrome_storage_local_set_sink: 676 cs_0.js, 677 cs_0.js

```

676: window.addEventListener("PassAccessToken", function (evt) { // VULNERABILITY LINE
677:   chrome.storage.local.set({ accessToken: evt.detail }); // VULNERABILITY LINE
678: }, false);

```

34. Lbcabedlbpobhpongobnfbkfceedgdjk TP

Sinks

cs_window_eventListener_message → chrome_storage_sync_set_sink: 535 cs_0.js

Minified code - no line specified by coco

```

window.addEventListener(
  "message",
  function (e) {
    var t = "chrome-extension://" + chrome.runtime.id;
    if (e.origin == t) {
      var n = u();
      if (!n) return;
    } else if (e.data.camerasOn) {
chrome.storage.sync.set({ enableEmbeddedCamera:e.data.camerasOn.toString() }),

```

35. Ljophmlbljnjodcbogmdogcpclifenpk no res.txt only res_old.txt TP

Cs send to BG

```

const port = chrome.runtime.connect();
port.onMessage.addListener(function(msg) {
  if (msg.type && msg.type == "crestron.airmedia.query.response") {
    window.postMessage(msg, "*");
  }
});

```

from cs_window_eventListener_message to localStorage_setItem_value

```

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

```

```

    if (event.data.type && (event.data.type ==
"crestron.airmedia.query.request" || event.data.type ==
"crestron.airmedia.connect.request")) {
        port.postMessage(event.data);
    }
}, false);

```

[bg.js](#): = Storage set =?

```

shareTo: function(a, c) {
    window.connectState = ConnectionState.CSConnecting;
    var d = a.code.split("?????");
    setPars(d[1]);
    storageSet(s_idx_ip, a.code);
}

```

36. Lonpbllihccdolpahofjieaklpckdmc

from storage_sync_get_source to sendResponseExternal_sink TP:

Get and set

```

h.tabs.onUpdated.addListener(v),
h.runtime.onMessageExternal.addListener((function(e, t, n) {
    switch (e.message) {
        case "setAnalyticsOption":
            ! function(e, t, n) {
                h.storage.sync.get("config", (function(t) {
                    var r = t.config || {},
                        i = "0";
                    e.decision && (i = "" + (new
Date).getTime()), r.analyticsOptIn = i, h.storage.sync.set({
                        config: r
                    }), n({
                        isSubmitted: !0
                    })
                })))
            }(e, 0, n);
            break;
    }
}

```

37. Mbpfpileganodbgnfdadpokoijibnfhjb

Sink 1: bg_chrome_runtime_MessageExternal → fetch_resource_sink + fetch_options_sink

TP fetch to attacker url

****Code Evidence**** (background.js:16-24):

```

chrome.runtime.onMessageExternal.addListener(function(message, sender,
sendResponse) {
    const char = message.auth.uri.match(/\?./) ? '&' : '?';
    const uri = message.auth.uri + char + 'version=' + EXTENSION_VERSION;
}

```

```

    fetchToJSONResponse(uri, message.auth.options).then(response => {
        if (!response.ok) return sendResponse(response)
        router(message.request, sender, sendResponse)
    })
    return true
});

```

38. Pdjmoeqkkmdbohppjcnpfcgajjniepn

Sink 1: bg_chrome_runtime_MessageExternal → BookmarkCreate_sink TP

Sink 2: bg_chrome_runtime_MessageExternal → localStorage_setItem_key/value TP

```

chrome.runtime.onMessageExternal.addListener(function(request, sender,
sendResponse) {
    if (request.checkStatus) {
        sendResponse({success: true});
    }
    else if (request.addBookmarkUrl) {
        chrome.bookmarks.create({
            title: request.title,
            url: request.addBookmarkUrl
        });
        localStorage.setItem(request.addBookmarkUrl,
request.bookCoverUrl);
        localStorage.setItem(request.addBookmarkUrl +
"::expired", request.expired);
        sendResponse({success: true});
    }
}

```

Sink 3: bg_chrome_runtime_MessageExternal → localStorage_remove_sink TP

```

    else if(request.deleteBookmarkUrl) {
        mThis.deleteBookmarks(request.deleteBookmarkUrl);
        localStorage.removeItem(request.deleteBookmarkUrl);
        sendResponse({success: true});
    }
}

```

39. phgddhgfnjjaobkeekohieahfingldac

Sink 1: document_eventListener_fdWebExtension.saveToStorage →

chrome_storage_local_set_sink

```

document.addEventListener('fdWebExtension.saveToStorage', e => {
    var key = e.detail.key;
    var value = e.detail.value;
    chrome.storage.local.set({ [key]: value });
});

```

Sink 2: document_eventListener_fdWebExtension.deleteFromStorage →
chrome_storage_local_remove_sink TP

```
document.addEventListener('fdWebExtension.deleteFromStorage', e => {  
  var key = e.detail.key;  
  chrome.storage.local.remove(key, () => {  
    if (chrome.runtime.lastError) {  
      console.error(chrome.runtime.lastError.message);  
    }  
  });  
});
```