

CoCo Verification:

### 1. adhfophjkhfbceaeljogehoikdnalgnrom - 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 sanitised - 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. aiacepmnmgokgjaplbbloiccdmmfcaa - 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 +
'//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. Bblkpdkdloalbiifhhmekiaejmdkgohj 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 xhttp = new XMLHttpRequest();
        xhttp.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. Bglneidhakmhndnbiggoldnkdgbcckdeck 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     }
1346
```

### - 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. Bojkjlencpogikokoooecajmflfobpcc 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 = 'bojkjlencpogikokoooecajmflfobpcc'
  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: sendResponse(JSON.stringify(items)) // VULNERABILITY LINE
8. 973:         })
9. 974:         break

```

## 9. Cdapnbiifmnajacjlfiikicefmidkbdl 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.injectedscrip !== '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. Cmadiiggcaaelekacljabmbefghaif 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. Cmlkmalcjcbmledhdedbljhfejciicbh 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.futalert.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.futalert.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. Cnaealcmncpabiiolgcmjnflpdalknjo 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://*.freeaddon.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.app GetAll) {
    chrome.management.getAll(function(e) {
        o(e)
    });
    return true
}

```

### 13. Cngodoeanflglpclhnlcohpgejhglfk 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, tabchgobj, 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. Djdfomqalgbgnhknemfkegbehojkmkii 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. Eggdmhdppfgikgakkfojgiledkefdce 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 .ejlghibhonddakjbgbfmdpeifcoljnlob 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() {
844:         for (let i = 0; i < params.length; i++) {
845:             const element = params[i];
846:             if ($(`element.elementtocheck`).length > 0) {
847:                 clearInterval(checkExist);
848:                 `#${element.pricechart_id}.remove();
849:                 if (element.insertbefore !== "") {
850:                     `#<div id='` + element.pricechart_id +
851: 'Lịch sử giá</div>`).insertBefore(element.insertbefore);
852:                 }
853:                 if (element.insertafter !== "") {
854:                     `#<div id='` + element.pricechart_id +
855: 'Lịch sử giá</div>`).insertAfter(element.insertafter);
856:                 }
857:                 if (element.tiki_api_product !== "") { //
```

VULNERABILITY LINE

```
858:                     $.ajax({
859:                         url: element.tiki_api_product,
860:                         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 snippet 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: " +
encodeURI(data.coordX) + ", coordY: " + encodeURI(data.coordY) + "
}, '*' );"});
        ...
    });
    return true;
});

```

## 18. Fmiojochalhealflohaicjncoofdjfb FP - contains no data

## 19. Foahoboefnlidmejppiibgnifpbjeknh TP

Foahoboefnlidmejppiibgnifpbjeknh 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. Foikfmnjgoljejophccdlhenbkogiemo FP no results

Error:/Users/jianjia/Documents/tmp/EOPG/result\_analyze/opgen\_results/server/all/detected/foikfmnj

## 21. Ggjakfijchdkbmmhbfbemjciidhnippgoe 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. GkcaldnjihkfdlegnanicInpdfcaid 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. GkofckdfjcapgbdIkifdlbkbbdpbgmfl 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

```
716:     function R(t) {
717:         for (var a = 0; a < e.storageDefaultKeys.length; a++) {
718:             var o = e.storageDefaultKeys[a];
719:             if (o === "enable_countdown") {
720:                 delete t.changeOptions["enable_countdown"]; // VULNERABILITY LINE
721:                 delete t.changeOptions["countdownPosition"]; // VULNERABILITY LINE
722:                 delete t.changeOptions["countdownText"]; // VULNERABILITY LINE
723:                 delete t.changeOptions["countdownToTime"]; // VULNERABILITY LINE
724:                 delete t.changeOptions["countdown_text_color"]; // VULNERABILITY LINE
725:                 delete t.changeOptions["countdown_background"]; // VULNERABILITY LINE
726:                 delete t.changeOptions["countdown_notified"] // VULNERABILITY LINE
727:             } else if (typeof t.changeOptions[o] !== "undefined") delete t.changeOptions[o]
// VULNERABILITY LINE
728:         }
729:         if (t.changeOptions["had_wl"]) { // VULNERABILITY LINE
730:             var l = JSON.parse(localStorage.getItem("had_wl")) || [];
731:             var r = utils.getAppsInList2ThatNotInList1(l,
JSON.parse(t.changeOptions["had_wl"]));
732:             if (r && r.length) {
733:                 if (e.debug) console.log("add to wl: ", r);
734:                 l = [].concat(l, r);
735:                 localStorage.setItem("had_wl", JSON.stringify(l))
736:             }
737:             delete t.changeOptions["had_wl"]
738:         }
739:         if (t.changeOptions.enable_most_visited) localStorage.setItem("enable_most_visited",
t.changeOptions.enable_most_visited); // VULNERABILITY LINE
740:         else if (t.changeOptions.disable_most_visited)
localStorage.setItem("enable_most_visited", t.changeOptions.disable_most_visited == "yes" ? "no" :
"yes"); // VULNERABILITY LINE
741:         if (t.changeOptions.enable_apps) localStorage.setItem("enable_apps",
t.changeOptions.enable_apps); // VULNERABILITY LINE
742:         else if (t.changeOptions.disable_apps) localStorage.setItem("enable_apps",
t.changeOptions.disable_apps == "yes" ? "no" : "yes"); // VULNERABILITY LINE
743:         if (t.changeOptions.enable_share) localStorage.setItem("enable_share",
t.changeOptions.enable_share); // VULNERABILITY LINE
744:         else if (t.changeOptions.disable_share) localStorage.setItem("enable_share",
t.changeOptions.disable_share == "yes" ? "no" : "yes"); // VULNERABILITY LINE
745:         if (t.changeOptions.enable_todo) localStorage.setItem("enable_todo",
t.changeOptions.enable_todo); // VULNERABILITY LINE
746:         else if (t.changeOptions.disable_todo) localStorage.setItem("enable_todo",
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. Hdbenfgeomkaiddadkoammkpeofpkj FP

message external set coookie, 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.sendNativeMessage(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.llecenhheemgphjfutmgnhllgcifpi FP

from bg\_chrome\_runtime\_MessageExternal to localStorage\_setItem\_value: 866  
set attacker and get to hardcoded url

```

var API_URL = 'https://app.letslyn.com/backend/plugin/web/';
LAUNCHER_URL = 'https://launcher.letslyn.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. lolIppockafcienapcpdpbmbdolbpgn 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. Jijnheciphcicdogdihjknnndkhagdn 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(bramNtr) {
    if (bramNtr.type === 'download') {
        frin(bramNtr.url, bramNtr.filename);
    } else if (bramNtr.type === 'open_link') {
        gabar(bramNtr.url)
    } else if (bramNtr.type === 'gabar') {
        gabar();
    }
}

function frin(tyNkot, nbaTyb) {
    chrome.downloads.download({
        url: tyNkot,
        filename: nbaTyb,
    });
}

```

### **29. jnhcnpjilgnklonkjpdamjghjbpiicao 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. Kafdgfnbckhdajclalkhoebpoppbogkd TP

!!! similar as Gkofckdfjcapgbdlkfdbbpbgmfl 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    "chrome_url_overrides": {
16      "newtab": "html/index.html"
17    },
18   "description": "Roblox extension provide you with HD Roblox wallpaper backgrounds. Themes designed for Roblox fans.",
19   "icons": {
20     "128": "html/skin/icons/logo_128.png",
21     "16": "html/skin/icons/logo_16.png",
22     "48": "html/skin/icons/logo_48.png"
23   },
24 },
25   "manifest_version": 2,
26   "name": "Roblox Download",
27   "offline_enabled": true,
28   "permissions": [ "topSites" ],
29 },
30   "version": "0.0.10.289.107",
31   "web_accessible_resources": [ "html/skin/**" ]
32 }
33
34
35
36 }
```

## 31. Kalfcfpaabnmkndhefnijjhcn djhni jc 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. Kankhbhekghfgkfbeafc ahgnpgnniajg 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 Kmjbjddieihpeglpobkhcnbejooilh 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. Lbcabedlbpobhpongobnfbfceedgdjk 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.cameralsoOn) {
      chrome.storage.sync.set({ enableEmbeddedCamera:e.data.cameralsoOn.toString() }),
    }
  }
);

```

### 35. Ljophmlbljnjobcbogmdogcpclifenpk 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 = ConnectState.CSConnecting;
    var d = a.code.split("?????");
    setPars(d[1]);
    storageSet(s_idx_ip, a.code);
}

```

### 36. Lonpjblilihcccdolpahofjiekaklpckdmc

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. Mbpfipleganodbgndfadpokojibnfhjb

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. Pdjmoegkkmdbohppjcnpfcggajjniepn

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. phgddhgfnijaobkeekohieahfingldac

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);
    }
  });
});
```