

Repackaging in Third-Party Android Marketplaces

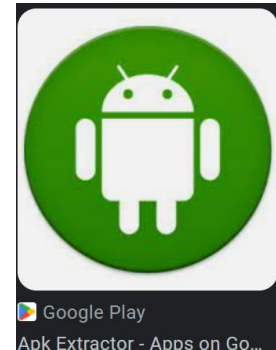
Anol Kurian Vadakkeparampil, Rohan Paranjpe, Neha Komuravelly

Collecting Data:

We get apk files for playstore using the gplaycli.

Sample : python GooglePlayCrawler.py -d DIR (-a [ALL] -c [COLLECTION] -g [CATEGORY] -n [NUMBER] -s [START])

We use an app available on playstore/f-droid to extract the apk directly from an android device. Since we only look at free apps we do not need root access for the same. Additionally, we also made sure to download apks of the same version.



Extracting Dex Codes:

We use androguard to fetch dex codes of extracted apks (pip install androguard). The file code7.py implements the logic to create dex files in batches of category folders. Update the folder paths as needed in order to get dex codes. We have generated approximately **100 GB** of dexcode data.

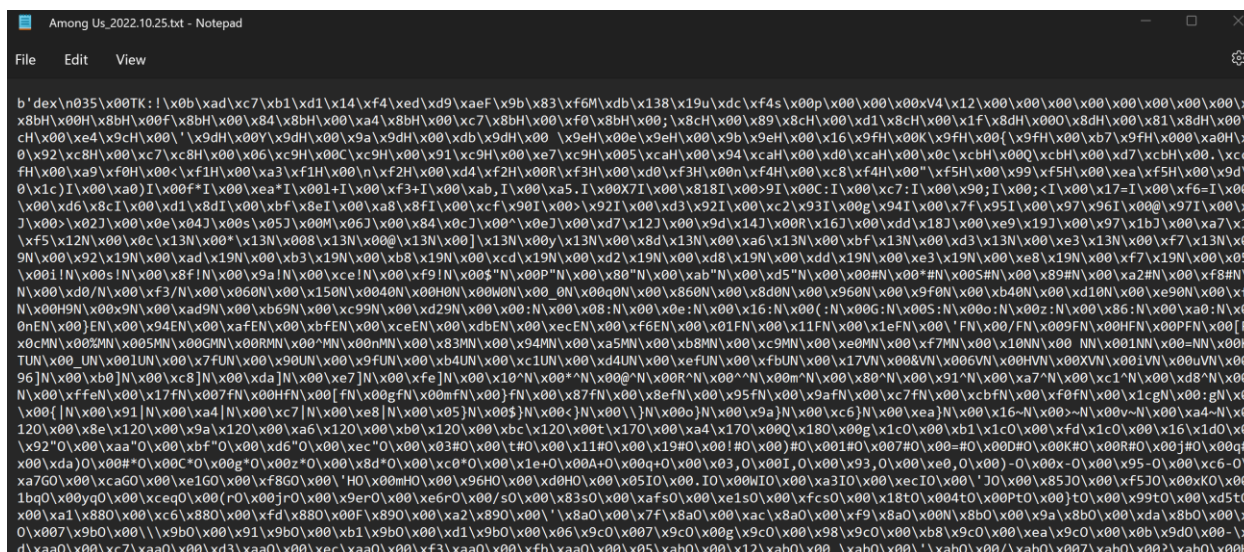
Generating Signatures:

We generate signatures using T-CTPH and generate a similarity score for the apks by comparing the generated signatures. The file hash.py contains the code to generate the T-CTPH. One can manipulate the file paths to generate their own hashes if needed. We use ppdeep as our base to build our own code to run T-CTPH. Hashing process takes approximately 0.85 minutes per MB to generate a signature.

Terminal Snippet of hash generation, comparison and similarity score generation:

```
hash2 is
['196608:J5A7Cfh/kPGdmSbaLDbJE+2+e0z78XyrN8RewEmpuSNUGhY0:0Cfh/sGdmSbaLdkOoyrN8Re45', '196608:1D+BiVrXIhg9W8LEpNFL0muw607LaHutreNycUOHC
GHwKn8U+:iVrX+g9W8LEpNiOyUtreNyc/']
hash2 is
['196608:bsK9TnLr0mmJY+8LhpKJ2exKnwezFn9eYA6C3:1XQ01w', '196608:XUGfNzHLo0iNSCmDfhoz4CPErM4FXBrMAie6v:LJIgzW']
true
Time 3: 589.2718805999612
hash3
['196608:bsK9TnLr0mmJY+8LhpKJ2exKnwezFn9eYA6C3:1XQ01w', '196608:XUGfNzHLo0iNSCmDfhoz4CPErM4FXBrMAie6v:LJIgzW']
true
Time 4: 646.1122793999966
hash4
['196608:bsK9TnL400mmJY+8LhpKJ2exKnwezFn9eYA6CB:1MmQ012', '196608:XUGfNzHGQo0iNSCmDfhoz4CPErM4FXBrMAie65:LaiIgzY']
Apkmirror repackaged app Alibaba.com_8.6.1.txt in category Shopping with similarity score : 97.0
D:/CNS/apk/playstore/Shopping/dex/Alibaba.com_8.6.1.txt
Time 1: 974.3155957999988
['196608:nQRW8eWm+6+fJwz3rPLmmwCn/7RUv6cCTyDeSxLi/7RUv6cCTyDRvxH6QR/7g:nz8eWhLfUwJXosfn4j0Mu', '196608:7KJIGwKr0mGQ5BU7IntfjEwiKN5LN
GpkOWh05IWLyx5SLNGpkOWh05zxpFqch5Le:7TGwKrr01dyui3LyeBrEHao6']
D:/CNS/apk/playstore/Shopping/dex/Alibaba.com_8.6.1.txt
Time 2: 975.6381767999846
hash2 is
['196608:nQRW8eWm+6+fJwz3rPLmmwCn/7RUv6cCTyDeSxLi/7RUv6cCTyDRvxH6QR/7g:nz8eWhLfUwJXosfn4j0Mu', '196608:7KJIGwKr0mGQ5BU7IntfjEwiKN5LN
GpkOWh05IWLyx5SLNGpkOWh05zxpFqch5Le:7TGwKrr01dyui3LyeBrEHao6']
true
file does not exist for hash 3
Time 4: 1208.285850699991
hash4
['196608:nQRW8eWm+6+UrpH9bJwz3rPLmmwCn/7RUv6cCTyDeSxLi/7RUv6cCTyDRvxHg:nz8eWhLsh9buUwJXosfn4j0Mu', '196608:7KJIGwKr0mGQKnn1dBU7Intfj
EwiKN5LNngpkOWh05IWLyx5SLNGpkOWh05zxpFqch5Le:7TGwKrr01dyui3LyeBrEHao6']
Apkmirror repackaged app AliExpress_8.58.0.txt in category Shopping with similarity score : 93.0
D:/CNS/apk/playstore/Shopping/dex/Amazon Shopping_24.21.4.100.txt
Time 1: 994.6739593000384
['196608:3h93wK0t8W6C8hFQ3fG5tCfL70tCD3v8tAF1AD0tGFLt64t1t6uJt373LstGwKrr01dyui3LyeBrEHao6', '196608:3h93wK0t8W6C8hFQ3fG5tCfL70tCD3v8tAF1AD0tGFLt64t1t6uJt373LstGwKrr01dyui3LyeBrEHao6']
```

DexCode Txt File Snippet:



Following is the file hierarchy we have implemented:

File Explorer window showing the contents of the 'apk' folder.

Name	Date modified	Type	Size
apkmirror	11/25/2022 5:41 PM	File folder	
Aptoid	11/27/2022 12:59 AM	File folder	
aurora	11/25/2022 5:40 PM	File folder	
playstore	11/25/2022 3:08 PM	File folder	

CNS > apk > apkmirror

Name	Date modified	Type	Size
Games	11/25/2022 6:45 PM	File folder	
Productivity	11/27/2022 12:59 AM	File folder	
Shopping	11/27/2022 2:48 PM	File folder	
Social	11/27/2022 12:57 AM	File folder	

CNS > apk > aurora > Games					Search Games	
Name	Date modified	Type	Size			
dex	12/5/2022 3:56 PM	File folder				
Among_Us_base.apk	11/26/2022 4:32 PM	APK File	9,274 KB			
Car_Crash_base.apk	11/26/2022 4:32 PM	APK File	81,949 KB			
check.txt	12/5/2022 4:27 PM	Text Document	1 KB			
Color_Page_base.apk	11/26/2022 4:32 PM	APK File	101,524 KB			
Coloring_Master_ASRM_base.apk	11/26/2022 4:32 PM	APK File	36,136 KB			
Door_Alphabet_base.apk	11/26/2022 4:32 PM	APK File	36,809 KB			
FIFA_Mobile_base.apk	11/26/2022 4:32 PM	APK File	94,194 KB			
Going_Balls_base.apk	11/26/2022 4:32 PM	APK File	168,941 KB			
Hunt_And_Seek_base.apk	11/26/2022 4:32 PM	APK File	57,106 KB			
Lords_Mobile_base.apk	11/26/2022 4:32 PM	APK File	67,531 KB			

CNS > apk > aurora > Games > dex					Search dex	
Name	Date modified	Type	Size			
Among Us_2022.10.25.txt	12/5/2022 3:56 PM	Text Document	21,307 KB			
Car Crash_1.9.txt	12/5/2022 3:57 PM	Text Document	73,164 KB			
Color Page_1.0.8.txt	12/5/2022 3:58 PM	Text Document	89,510 KB			
Coloring Master ASRM_1.20.txt	12/5/2022 3:59 PM	Text Document	76,932 KB			
Door Alphabet_0.5.txt	12/5/2022 4:00 PM	Text Document	86,327 KB			
FIFA Mobile_18.0.02.txt	12/5/2022 4:02 PM	Text Document	77,375 KB			
Going Balls_1.39.txt	12/5/2022 4:03 PM	Text Document	101,132 KB			
Hunt & Seek_1.0.8.txt	12/5/2022 4:04 PM	Text Document	92,498 KB			

Code Snippet:

```
directory_in_str = "D:/CNS/" + folder + "/playstore/" + category + "/dex/"
directory = os.fsencode(directory_in_str)
for file in os.listdir(directory):
    filename = os.fsdecode(file)
    print("D:/CNS/" + folder + "/playstore/" + category + "/dex/" + filename)
    start = timeit.default_timer()
    hash1 = hash_from_file("D:/CNS/" + folder + "/playstore/" + category + "/dex/" + filename)
    stop = timeit.default_timer()
    print('Time 1: ', stop - start)
    print(hash1)
    print("D:/CNS/" + folder + "/playstore/" + category + "/dex/" + filename)
    if os.path.isfile("D:/CNS/" + folder + "/aurora/" + category + "/dex/" + filename):
        start = timeit.default_timer()
        hash2 = hash_from_file("D:/CNS/" + folder + "/aurora/" + category + "/dex/" + filename)
        stop = timeit.default_timer()
        print('Time 2: ', stop - start)
        print("hash2 is " )
        print(hash2)
        sum2 = 0
        if(hash2 == hash1):
            print("true")
        else :
            for x, y in zip(hash1, hash2):
                sum2 = sum2 + compare(x,y)
            tempstr = "Aurora repackaged app "+filename+" in category " + category+ " with similarity score "
            print("Aurora repackaged app with similarity score : " + sum2 )
            with open("D:/CNS/Output.txt", 'a') as f:
                f.write(str(tempstr) + "\n")
    else:
        print("file does not exist for hash 2")

#### ----BELOW CODE IS DUPLICATE OF ABOVE LOGIC WITH PATHS CHANGED FOR RESPECTIVE DIRECTORIES ---- ####
```

```
for file in os.listdir(directory):
    filename = os.fsdecode(file)
    if filename.endswith(".apk"):
        d[filename]=0
json.dump(d, open(directory_in_str+"check.txt", 'w'))
d2 = json.load(open(directory_in_str+"check.txt"))
try:
    print("here3")
    print(directory)
    for file in os.listdir(directory):
        filename = os.fsdecode(file)
        print(filename)
        if filename.endswith(".apk"):
            print(filename)
            print(d2[filename])
            if d2[filename] != 1:
                a, d, dx = AnalyzeAPK(directory_in_str+filename)
                version = a.get_androidversion_name()
                txtfilename = a.get_app_name()
                print("here3")
                if not os.path.exists(directory_in_str+'dex/'+txtfilename+'_'+version+'.txt'):
                    getalldex = a.get_all_dex()
                    with open(directory_in_str+'dex/'+txtfilename+'_'+version+'.txt', 'w') as f:
                        for x in getalldex:
                            f.write(str(x))
                d2[filename]=1
            continue
    json.dump(d2, open(directory_in_str+"check.txt", 'w'))
except Exception as e:
    print(e)
    json.dump(d2, open(directory_in_str+"check.txt", 'w'))
```