APK	ID	Source Signature	Source Category	Sink Signature	Sink Category	Description
						This malicious flow leaks the SIM country code into an OutputStream, writing to data that is later read by
backflash	1	android.telephony.TelephonyManager: java.lang.String getSimCountryIso()	LOCATION INFORMATION	java.io.OutputStreamWriter: void write(java.lang.String)	FILE	following flow.
		android.content.Context: java.io.FileInputStream		java.net.HttpURLConnection: java.io.OutputStream		The second part of the above partial flow that leaks the country code into a URL that is then retrieved.
backflash	2	openFileInput(java.lang.String)	FILE	getOutputStream()	NETWORK	Note: execute() internally calls doInBackground().
h l-fl h	_	and and another than the second and	CONTACTINEODNAATION	in a la Data Outra (Changa and Andrew Line)	NETHODIC	This walk is a first of the same dark and also have a first of the same dark and a same dark and a same dark a
backflash	3	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.DataOutputStream: void write(byte[],int,int) java.net.HttpURLConnection: java.io.OutputStream	NETWORK	This malicious flow sends the telephone number of incoming calls via file upload to a remote server.  This malicious flow reads incoming number and appends it in a URL as parameter for a HTTP POST
backflash	4	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	getOutputStream()	NETWORK	request to a remote server.
Duckingsii	-	and outcome intent. Javanang. String getstring. Extragavariang. String)	CONTACT IN CHIMATION	getoutputstream()	NETWORK	request to diffinite server.
backflash	5	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.OutputStreamWriter: void write(java.lang.String)	NETWORK	This malicious flow writes phone number to a file which will be posted to a remote server.
backflash	6	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.OutputStreamWriter: void write(java.lang.String)	NETWORK	This malicious flow writes phone number to a file which will be posted to a remote server.
	_					
backflash	7	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.OutputStreamWriter: void write(java.lang.String)	NETWORK	This malicious flow writes phone number to a file which will be posted to a remote server.
backflash	8	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.OutputStreamWriter: void write(java.lang.String)	NETWORK	This malicious flow writes phone number to a file which will be posted to a remote server.
Dackilasii	0	and old.content.intent. Java.iang.string getstringextratjava.iang.string)	CONTACT INFORMATION	Java.io.Outputstreamwriter. void writetjava.iang.string)	INETWORK	This manicious now writes priorie number to a file which will be posted to a remote server.
backflash	9	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.OutputStreamWriter: void write(java.lang.String)	NETWORK	Negative flow.
		,		,		
backflash	10	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.OutputStreamWriter: void write(java.lang.String)	NETWORK	Negative flow.
				java.net.HttpURLConnection: java.io.OutputStream		
backflash	11	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	getOutputStream()	NETWORK	Negative flow.
				java.net.HttpURLConnection: java.io.OutputStream		
backflash	12	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	getOutputStream()	NETWORK	Negative flow.
backflash	13	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.net.HttpURLConnection: java.io.OutputStream getOutputStream()	NETWORK	Negative flow.
Duckingsii	13	and outcome intent. Javanang. String getstring. Extragavanang. String)	CONTACT IN CHIMATION	java.net.HttpURLConnection: java.io.OutputStream	NETWORK	regulive now.
backflash	14	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	getOutputStream()	NETWORK	Negative flow.
		, , , , , , , , , , , , , , , , , , , ,		java.net.HttpURLConnection: java.io.OutputStream		
backflash	15	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	getOutputStream()	NETWORK	Negative flow.
				java.net.HttpURLConnection: java.io.OutputStream		
backflash	16	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	getOutputStream()	NETWORK	Negative flow.
backflash	17	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.OutputStreamWriter: void write(java.lang.String)	NETWORK	This malicious flow writes incoming phone number to a file which will be posted to a remote server.
Dackilasii	17	and old. Content. Intent. Java.iang. String getStringExtratjava.iang. String)	CONTACT INFORMATION	Java.io.Outputstreamwriter. void writetjava.iang.string)	INETWORK	This manicious now writes incoming priorie number to a nie which will be posted to a remote server.
backflash	18	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.OutputStreamWriter: void write(java.lang.String)	NETWORK	Negative flow.
		g		)ggg		
backflash	19	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.DataOutputStream: void write(byte[],int,int)	NETWORK	This malicious flow writes phone number to a file and the file content is posted to a remote server.
backflash	20	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.DataOutputStream: void write(byte[],int,int)	NETWORK	This malicious flow writes phone number to a file and the file content is posted to a remote server.
	L.					
backflash	21	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.DataOutputStream: void write(byte[],int,int)	NETWORK	This malicious flow writes phone number to a file and the file content is posted to a remote server.
backflash	22	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.DataOutputStream: void write(byte[],int,int)	NETWORK	This malicious flow writes phone number to a file and the file content is posted to a remote server.
Duckingsii		and outcome intent. Javanang. String getstring. Extragavariang. String)	CONTACT IN CHIMATION	java.io.batao atputstream. voia write(byte(j,int,int)	NETWORK	This manifolds now writes phone number to a file and the file content is posted to a remote server.
backflash	23	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.DataOutputStream: void write(byte[],int,int)	NETWORK	Negative flow
backflash	24	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	java.io.DataOutputStream: void write(byte[],int,int)	NETWORK	Negative flow
		android.content.ContentResolver: android.database.Cursor				
beita_com_beit		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav		javax.mail.Transport: void		
a_contact		a.lang.String)	DATABASE	sendMessage(javax.mail.Message,javax.mail.Address[])	SMS MMS	This malicious flow leaks contact data via email.
beita com beit		android.content.ContentResolver: android.database.Cursor query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				
		a.lang.String)	DATABASE	java.io.BufferedWriter: void write(java.lang.String)	FILE	This malicious partial flow leaks contact data into a file (that is later uploaded).
beita com beit	_			January State Control of the Control		This malicious partial flow uploads the file containing contact data to a remote server via HTTP POST
	3	java.io.FileInputStream: void <init< td=""><td>FILE</td><td>java.io.DataOutputStream: void write(byte[],int,int)</td><td>NETWORK</td><td>request.</td></init<>	FILE	java.io.DataOutputStream: void write(byte[],int,int)	NETWORK	request.
				com.baidu.inf.iis.bcs.BaiduBCS:		This malicious flow reads photos and uploads them to a remote server via HTTP request. This malicious
1				com.baidu.inf.iis.bcs.response.BaiduBCSResponse		behavior will be triggered, when then PushMessageReceiver (a BroadcastReceiver) receives a command
cajino baidu	1	java.io.File: java.io.File[] listFiles()	FILE	putObject(com.baidu.inf.iis.bcs.request.PutObjectRequest)	NETWORK	which contains the string 'photo'.

						This malicious flow reads contacts and saves them to a file. This file will be later uploaded to a remote
						server (not a part of this flow). This malicious behavior will be triggered, when then
cajino_baidu	2	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: void write(java.lang.String)	FILE	PushMessageReceiver (a BroadcastReceiver) receives a command which contains the string 'contact'.
						This malicious flow reads call log and saves them to a file. This file will be later uploaded to a remote
cajino baidu	3	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: void write(java.lang.String)	FILE	server (not a part of this flow). This malicious behavior will be triggered, when then  PushMessageReceiver (a BroadcastReceiver) receives a command which contains the string 'call log'.
cajirio_baidu	3	and old. database. Cursor. Java.lang. String getstring(int)	DATABASE	java.io.i nevvitei. void writetjava.iang.5timg)	TILL	This malicious flow reads sms messages and saves them to a file. This file will be later uploaded to a
						remote server (not a part of this flow). This malicious behavior will be triggered, when then
						PushMessageReceiver (a BroadcastReceiver) receives a command which contains the string
cajino_baidu	4	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: void write(java.lang.String)	FILE	'upload_message'.
		andraid leastion Leastion Managary andraid leastion Leastion				This malicious flow reads location information and saves it to a file. This file will be later uploaded to a
cajino_baidu	5	android.location.LocationManager: android.location.Location getLastKnownLocation(java.lang.String)	LOCATION INFORMATION	java.io.FileWriter: void write(java.lang.String)	FILE	remote server (not a part of this flow). This malicious behavior will be triggered, when then PushMessageReceiver (a BroadcastReceiver) receives a command which contains the string 'location'.
cajiilo_balaa	-	gettastanowntocationiquva.iang.string/	LOCATION IN CHIMATION	java.io.i ne vvitter. void write (java.iung.string)	TIEL	a distinct stage receiver (a broadcastreceiver) receives a command which contains the string location.
						This malicious flow reads device Id and saves it to a file. This file will be later uploaded to a remote
						server (not a part of this flow). This malicious behavior will be triggered, when then
cajino_baidu	6	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	java.io.FileWriter: void write(java.lang.String)	FILE	PushMessageReceiver (a BroadcastReceiver) receives a command which contains the string 'phone'.
				com.baidu.inf.iis.bcs.BaiduBCS:		This malicious flow reads device Id and uploads it to a remote server via HTTP request. This malicious
cajino baidu	_	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	com.baidu.inf.iis.bcs.response.BaiduBCSResponse putObject(com.baidu.inf.iis.bcs.request.PutObjectRequest)	NETWORK	behavior will be triggered, when then PushMessageReceiver (a BroadcastReceiver) receives a command which contains the string 'phone'.
cajiiio_baidu	-	and out telephony. relephonyivianager, java.iang.string gerbeviceid()	UNIQUE IDENTIFIER	putobject(com.baidd.im.iis.bcs.request.rutobjectkequest)	INETWORK	which contains the string phone .
						This malicious flow reads file list and saves it to a file. This file will be later uploaded to a remote server
						(not a part of this flow). This malicious behavior will be triggered, when then PushMessageReceiver (a
cajino_baidu	8	java.io.File: java.io.File[] listFiles()	FILE	java.io.FileWriter: void write(java.lang.String)	FILE	BroadcastReceiver) receives a command which contains the string 'list_file'.
				com.baidu.inf.iis.bcs.BaiduBCS:		
			5.1.5	com.baidu.inf.iis.bcs.response.BaiduBCSResponse	NETHORY	This malicious flow uploads files. This malicious behavior will be triggered, when then
cajino_baidu	9	java.io.File: void <init< td=""><td>FILE</td><td>putObject(com.baidu.inf.iis.bcs.request.PutObjectRequest)</td><td>NETWORK</td><td>PushMessageReceiver (a BroadcastReceiver) receives a command which contains the string 'upload_file'.</td></init<>	FILE	putObject(com.baidu.inf.iis.bcs.request.PutObjectRequest)	NETWORK	PushMessageReceiver (a BroadcastReceiver) receives a command which contains the string 'upload_file'.
						This malicious flow delete file. This malicious behavior will be triggered, when then
cajino_baidu	10	java.io.File: void <init< td=""><td>FILE</td><td>java.io.File: boolean delete()</td><td>CRITICAL FUNCTION</td><td>PushMessageReceiver (a BroadcastReceiver) receives a command which contains the string 'delete file'.</td></init<>	FILE	java.io.File: boolean delete()	CRITICAL FUNCTION	PushMessageReceiver (a BroadcastReceiver) receives a command which contains the string 'delete file'.
				com.baidu.inf.iis.bcs.BaiduBCS:		
				com.baidu.inf.iis.bcs.response.BaiduBCSResponse		
cajino_baidu	11	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	putObject(com.baidu.inf.iis.bcs.request.PutObjectRequest)	NETWORK	Negative flow. The device id is only used as folder name.
				com.baidu.inf.iis.bcs.BaiduBCS:		
cajino baidu	12	java.io.File: void <init< td=""><td>FILE</td><td>com.baidu.inf.iis.bcs.response.BaiduBCSResponse putObject(com.baidu.inf.iis.bcs.reguest.PutObjectReguest)</td><td>NETWORK</td><td>This malicious flow uploads files to a remote server.</td></init<>	FILE	com.baidu.inf.iis.bcs.response.BaiduBCSResponse putObject(com.baidu.inf.iis.bcs.reguest.PutObjectReguest)	NETWORK	This malicious flow uploads files to a remote server.
cajirio_baidu	12	Java.io.i iie. void viiit	ITEL	com.baidu.inf.iis.bcs.BaiduBCS:	INLIWORK	This mancious now uploads mes to a remote server.
				com.baidu.inf.iis.bcs.response.BaiduBCSResponse		
cajino_baidu	13	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	putObject(com.baidu.inf.iis.bcs.request.PutObjectRequest)	NETWORK	Negative flow. The device id is only used as folder name.
				com.baidu.inf.iis.bcs.BaiduBCS:		
				com.baidu.inf.iis.bcs.response.BaiduBCSResponse		
cajino_baidu	14	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	putObject(com.baidu.inf.iis.bcs.request.PutObjectRequest)	NETWORK	Negative flow. The device id is only used as folder name.
cajino_baidu	15	java.io.File: void <init android.content.context:="" java.io.fileinputstream<="" td=""><td>FILE</td><td>java.io.File: boolean delete()</td><td>CRITICAL FUNCTION</td><td>This malicious flow deletes files to cover up the malicious behaviors.</td></init>	FILE	java.io.File: boolean delete()	CRITICAL FUNCTION	This malicious flow deletes files to cover up the malicious behaviors.
chat hook	1	openFileInput(java.lang.String)	FILE	java.io.DataOutputStream: void writeBytes(java.lang.String)	CRITICAL FUNCTION	This malicious flow reads mount data to construct a command and execute it.
<u> </u>	1			org.apache.http.impl.client.DefaultHttpClient:		
				org.apache.http.HttpResponse		
chat_hook	2	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious partial flow reads chat history and sends it to a remote server per an HTTP POST request.
chat_hook	3	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs device id.
shot book	4	andraid contact Intent, involves Chrise setStringFutvo(inva long Chrise)	OTHER DATA	andraid still are intiliara long Chring into long Chring)	100	This malisians flour large shot history
chat_hook	4	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs chat history.
chat hook	5	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs last message time in chat history.
_		2 23 3 3 3 4 4				,
chat_hook	6	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs chat history.
chat_hook	7	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs imei.
chat hook	8	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs chat history.
chat_nook	0	and old content antent java lang string getstring Little (java lang string)	OTHER DATA	android.den.cog. int iljava.iang.string,java.iang.stringj	1200	mis manerous new rogs criat mistory.

and BankEndActivity, it starts from reading bank account number from user input in BankActivity, the user performs a click on a button, the BankNumActivity is started; the BankScativity and also started by clicking on a button. In the first three such as account number, passwords and TAN number are stored into the static fields of the class BankEndActivity are also started by clicking on a button. In the first three such as account number, passwords and TAN number are stored into the static fields of the class BankEnfor. There are input validations for user inputs to make sure that only valid bank information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInformation or apache. http.client.methods. HttpUriRequest)  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This is desamble. The propert is a start by reading be seried to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a list. This is desamble		T					
de took 12 anderdoorser, freste tijne sie jack programme in de production in tree de la million de la million fine à la	chat_hook	9	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs lat message time and with whom.
duty voice  12 and/ord content intent upon large performance intent group large performance i	chat_hook	10	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs imei.
due just 3 montal catenet serial pointing foreign price large (frequency of the pointing of th	chat_hook	11	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow is an implicit flow, since reaching sink depends on the data from source.
deposition content Connections designed per original per security and per 15 mg per	chat_hook	12	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs package information
dula 2   Lay String   Oxford And Association of the process of the	chat_hook	13	7 0 00 0 0 0	OTHER DATA	0 0 00 00	LOG	Negative flow.
Surface Strategy Control of the Surface Strategy Control of th					, , , , , , , , , , , , , , , , , , , ,		
Outland Souther Location Content Security Content Content S			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				
and the content content content deather printing printing printing and the content con	chulia	1	a.lang.String)	DATABASE		NETWORK	This malicious flow leaks the recipient or sender of an sms text message and posts it to an HTTP Server
substance of the control contr							
el monto content Content Recuber un monto di acabase Curso o grapache intro price de la chiefat Stription de content de la conte							
dest. from more and control control recording and process of the prophenome of the prophenome of the process of	chulia	2	0 0 0	LOCATION INFORMATION		NETWORK	This malicious flow leaks the location of the android device and posts it to an HTTP Server
Subject of the subjec							
undout content content content content content content and database curror dusting a lang String)  Out AgASE  Out a pasch this participate content content content content and starts as it independent on a content content and starts as it independent on a content content and starts as it independent on a content content and starts as it independent on a content content and starts as it independent on a content content and starts as it independent on a content and starts as it independent and starts as it independent on a content and starts as it independent and starts							
out-planed out-planed ane String Laws Jams String Laws Jams String out-planed and state strings are also stated by the plane partners of the state of the power number of a state of the power number of the po	chulia	3	0 0,	DATABASE		NETWORK	This malicious flow leaks contacts and posts it to an HTTP server.
Available					1 - 1		
android cfeephony gam. Smakharager: void denotype; and picture planes and provided seephone gam. Smakharager: void denotype; and picture planes and provided seephone gam. Smakharager: void denotype; and picture planes and provided seephone gam. Smakharager: void denotype; and picture planes and provided seephone gam. Smakharager: void denotype; and picture planes and provided seephone gam. Smakharager: void denotype; and picture planes and provided seephone gam. Smakharager: void denotype; and picture planes and provided seephone gam. Smakharager: void denotype; and picture planes and provided seephone gam. Smakharager: void denotype; and picture planes and provided seephone gam. Smakharager: void denotype; and picture planes and provided seephone gam. Smakharager: void denotype; and picture planes and provided seephone gam. Smakharager: void denotype; and provided seephone gam. Smakharager: void denotype gam. Sm					, , , ,		· ·
Seath File   Sea	chulia	4	a.lang.String)	DATABASE		NETWORK	AlarmService class, later puts this information into a intent and starts a BroadcastReceiver sendReceiver.
enaflow    1   extripsychessagebody    SMS MMS   roid app Penningintent android app Penninginten							
and concrete, Ex-Asset Manager java lang String per land							
Semonypersymen   1   Open (layour lang String)   FIE   (mixed (layour lang String)   CHITCAL FUNCTION   Contains malicious code is called via reflection.	erialflow	1	0 1 7 0 70	SMS MMS		SMS MMS	9 9
expression 1 android telephony TelephonyManager: java lang. String gettuen Number() UNIQUE IDENTIFER or gapache http://micro. Http://irequest) NETWORK This malicious flow leaks the telephone number to a remote server.  org. apache http://micro. Http://irequest) NETWORK This malicious flow leaks the telephone number to a remote server.  org. apache http://micro. Http://irequest) NETWORK This malicious flow leaks the telephone number to a remote server.  org. apache http://micro. Http://mi							
eyerspasm 1 android selephony. TelephonyManager java. Jang. String gettime Number() UNQUE IDENTIFIER or space-th. http://imet.cet. http://imet	dsencrypt_samp	1	open(java.lang.String)	FILE	invoke(java.lang.Object,java.lang.Object[])	CRITICAL FUNCTION	contains malicious code. The malicious code is called via reflection.
a product steephony, Telephony,							
Infinite Assistative Properties   First Assistative Properties   First Assistative Properties   First Assistation   First As							
manage@ueyinadroid.net.Urijava.lang.Stringjjava.lang.String.jjava.	exprespam	1		UNIQUE IDENTIFIER		NETWORK	This malicious flow leaks the telephone number to a remote server.
Expersion   2   Finish   Expersion   2   Finish   Expersion   Ex							
android content. ContentResolver: android database. Cursor queryIndroid one. Utilizava Jang String[,] ava Ja			managedQuery(android.net.Uri,java.lang.String[],java.lang.String,java.lang.St				
fakeappstore 1 a.lang.String	exprespam	2	ring[],java.lang.String)	DATABASE	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow leaks contact data to a remote server.
Seebank_andro   dasamp   1   fakebank_andro   dasamp   2   and ord. telephony. Telephony. Manager: Java Jang. String getDeviced()   UNIQUE IDENTIFIER   Org. apache. http. H					, , ,		
android telephony. Tel							
sendTextMessage(java.lang, String, java.lang, String, and roid app. Pendingintent, android app. Pendingintent, and	fakeappstore	1	a.lang.String)	DATABASE	11 0 1 11 0 1	SMS MMS	This malicious flow reads contacts from the device and spams them with SMS.
Fakeappstore 2 org.apache.http.HttpEntlty: java.io.inputStream getContent() INTERNET SOURCE rold.app.Pendingintent,android.app.Pendingintent,android.app.Pendingintent) SMS MMS (command and control).    Fakeappstore   3   android.telephony.TelephonyManager: java.lang.String getDeviceId()   UNIQUE IDENTIFIER   vex.cute(org.apache.http.client.methods.HttpUriRequest)   NETWORK   This malicious flow leaks the device id to an HTTP Server.    This malicious flow leaks the device id to an HTTP Server.							
fakeappstore 3 android.telephony,TelephonyManager: java.lang.String getDeviceId() UNIQUE IDENTIFIER org.apache.http.client.HttpClient: org.apache.http.client.httpClient.org.apa							9
fakeank_andro id_samp 1 android.telephony.TelephonyManager: java.lang.String getDeviceld() UNIQUE IDENTIFIER execute(org.apache.http.client.methods.HttpUriRequest) NETWORK This malicious ICC flow crosses 4 different activities: BankActivity, BankNumActivity BankScardAct and BankEndActivity, are also started; by client.methods.HttpUriRequest) This malicious ICC flow crosses 4 different activities: BankActivity, and BankEndActivity, are also started; by client.methods.HttpUriRequest) This malicious ICC flow crosses 4 different activities: BankActivity, and BankEndActivity are also started; by client.methods.HttpUriRequest) the user performs a facility on a button, the BankNumActivity is started; the BankScardActivity and BankEndActivity are also started by clicitor, on a button, the BankNumActivity is started; the BankScardActivity and BankEndActivity are also started by clicitor, on a button, the BankNumActivity is started; the BankScardActivity and BankEndActivity are also started by clicitor, on a button, the BankNumActivity is started; the BankScardActivity and BankEndActivity are also started by clicitor, on a button, the BankNumActivity is started; the BankScardActivity and BankEndActivity are also started; the BankScardActivity and BankEndActivity and BankEndActivity are also started; the BankScardActivity and BankEndActivity and BankEndAc	fakeappstore	2	org.apache.http.HttpEntity: java.io.InputStream getContent()	INTERNET SOURCE	roid.app.PendingIntent,android.app.PendingIntent)	SMS MMS	(command and control).
fakeappstore 3 android.telephony,TelephonyManager: java.lang,String getDeviceId() UNIQUE IDENTIFIER execute(org.apache.http.client.methods.HttpUriRequest) NETWORK This malicious ICC flow crosses 4 different activities: BankActivity, BankNumActivity BankScardAct and BankEndActivity are also started; the BankNumActivity and BankEndActivity are also started; the BankNumActivity and the sark activities. BankNumActivity are also started; the BankNumActivity and bank Information org. apache.http.HittpResponse and the class and the							
This malicious ICC flow crosses 4 different activities: BankActivity, BankScardActivity and BankEndActivity, It starts from reading bank account number from user input in BankActivity and BankEndActivity, the user performs a click on a button, the BankNumActivity is started; the BankScardActivity and BankEndActivity are also started by clicking on a button. In the first three activities, bank information org. apache.http.impl.client.DefaultHttpClient: org.apache.http.impl.client.DefaultHttpClient: org.apache.http.HttpResponse  daebank_andro id_samp  1 findViewByld(Int)  ACCOUNT INFORMATION							
and BankEndActivity, it starts from reading bank account number from user input in BankActivity, the user performs a click on a button, the BankNumActivity is attend; the BankScartavity and BankEndActivity are also started by clicking on a button. In the first three such as started; the BankScartavity and BankEndActivity are also started by clicking on a button. In the first three such as started; the BankScartavity and BankEndActivity are also started by clicking on a button. In the first three such as started; the BankScartavity and BankEndActivity are also started by clicking on a button. In the first three such is started; the BankScartavity and BankEndActivity are also started by clicking on a button. In the first three such is started; the BankScartavity and BankEndActivity, and collected in the last activities, and row in the first three such is started; the BankScartavity and BankEndActivity, and collected into the static fields of the class BankInfo. There are input validations for user inputs to make sure that only valid bank information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a collected. In the last activity, all collected information will be read from BankInfo and written to a starting bank activity, and collected in the last activity, all collected information will be read from BankInfo and written to a starting bank activity, and collected. In the last activity, all collected in	fakeappstore	3	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow leaks the device id to an HTTP Server.
id_samp 1 findViewByld(int) ACCOUNT INFORMATION execute(org.apache.http.client.methods.HttpUriRequest) NETWORK This list will be sent to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to the field of class BankGactivity, thento a HashMap will be appended to id_samp 2 getSimSerialNumber() UNIQUE IDENTIFIER java.net.HttpURIConnection: java.io.lnputStream getInputStream() NETWORK String which will be sent to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to the field of class BankGactivity, thento a HashMap will be appended to id_samp and roid.telephony.TelephonyManager: java.lang.String getSimSerialNumber() UNIQUE IDENTIFIER org.apache.http.client.DefaultHttpClient:  org.apache.http.client.methods.HttpUriRequest) NETWORK be sent to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This identification in the HashMap will be appended to id_samp and read to org.apache.http.client.methods.HttpUriRequest) NETWORK be sent to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This identifies the serial number of the SIM card, then writes to a List. This identifies the serial number of the SIM card, then writes to a BasicNameValuePair object, the server per an HTTP POST request.  fakebank_andro id_samp 4 android.telephony.SmsMessage: java.lang.String getDisplayMessageBody() SMS MMS  execute(org.apache.http.client.methods.HttpUriRequest) NETWORK by the class smsReceiver runs.							BankEndActivity are also started by clicking on a button. In the first three activities, bank information such as account number, passwords and TAN number are stored into the static fields of the class BankInfo. There are input validations for user inputs to make sure that only valid bank information is
fakebank_andro id_samp 2 getSimSerialNumber() UNIQUE IDENTIFIER UNIQUE IDENTIFIER UNIQUE IDENTIFIER org.apache.http.client.befaultHttpClient: org.apache.http.client.befaultHttpClient.befaultHt							
fakebank_andro id_samp 2 android.telephony.TelephonyManager: java.lang.String getSimSerialNumber() UNIQUE IDENTIFIER java.net.HttpURLConnection: java.io.InputStream getInputStream() NETWORK String which will be sent to a remote server per an HTTP GET request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This less than a more described by the class smsReceiver per an HTTP POST request when a Thread of a moderoid.telephony.SmsMessage: java.lang.String getDisplayMessageBody() SMS MMS  class BankEndActivity, thento a HashMap. Then information in the HashMap will be appended to String which will be sent to a remote server per an HTTP GET request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This less to a remote server per an HTTP POST request.  This malicious flow starts by reading SMS modern and the server per an HTTP POST request.  This malicious flow starts by reading SMS modern and the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when a Thread of the server per an HTTP POST request when	id_samp	1	findViewByld(int)	ACCOUNT INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	·
id_samp 2 getSimSerialNumber() UNIQUE IDENTIFIER java.net.HttpURLConnection: java.io.InputStream getInputStream() NETWORK String which will be sent to a remote server per an HTTP GET request.  org.apache.http.lient.DefaultHttpClient: org.apache.http.HttpResponse id_samp 3 getSimSerialNumber() UNIQUE IDENTIFIER  fakebank_andro id_samp 4 android.telephony.SmsMessage: java.lang.String getDisplayMessageBody() SMS MMS  UNIQUE IDENTIFIER  java.net.HttpURLconnection: java.io.InputStream getInputStream() NETWORK  org.apache.http.lient.DefaultHttpClient: org.apache.http.lient.methods.HttpUriRequest) NETWORK  NETWORK  String which will be sent to a remote server per an HTTP GET request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This leads to a remote server per an HTTP POST request.  NETWORK  fakebank_andro id_samp 4 android.telephony.SmsMessage: java.lang.String getDisplayMessageBody() SMS MMS  execute(org.apache.http.client.methods.HttpUriRequest) NETWORK  NETWORK  String which will be sent to a remote server per an HTTP GET request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This leads to a remote server per an HTTP POST request.  NETWORK  NETWORK  NETWORK  String which will be sent to a remote server per an HTTP GET request.							This malicious flow starts by reading the serial number of the SIM card, then writes to the field of the
org.apache.http.limpl.client. DefaultHttpClient: org.apache.http.limpl.client.DefaultHttpClient: org.apache.http.limpl.client.DefaultHttpClient: org.apache.http.HttpResponse id_samp 3 getSimSerialNumber()  UNIQUE IDENTIFIER  org.apache.http.client.methods.HttpUriRequest)  NETWORK  be sent to a remote server per an HTTP POST request.  fakebank_andro id_samp 4 android.telephony.SmsMessage: java.lang.String getDisplayMessageBody()  SMS MMS  org.apache.http.client.tethcdient.org.apache.http.HttpResponse  org.apache.http.client.tethcdient.org.apache.http.HttpResponse  execute(org.apache.http.Litent.methods.HttpUriRequest)  NETWORK  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This leads to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This leads to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This leads to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This leads to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This leads to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This leads to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This leads to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This leads to a remote server per an HTTP POST request.  This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This leads to a remote server per an HTTP POST request.	_						class BankEndActivity, thento a HashMap. Then information in the HashMap will be appended to a
fakebank_andro id_samp 3 android.telephony.TelephonyManager: java.lang.String getSimSerialNumber() UNIQUE IDENTIFIER org.apache.http.client.methods.HttpUriRequest) NETWORK be sent to a remote server per an HTTP POST request, then writes to a List. This is execute(org.apache.http.client.methods.HttpUriRequest) be sent to a remote server per an HTTP POST request, then writes to a BasicNameValuePair object, that is nationally an android.telephony.SmsMessage: java.lang.String getDisplayMessageBody() SMS MMS execute(org.apache.http.client.methods.HttpUriRequest) NETWORK by the class smsReceiver runs.	id_samp	2	getSimSerialNumber()	UNIQUE IDENTIFIER		NETWORK	String which will be sent to a remote server per an HTTP GET request.
id_samp 3 getSimSerialNumber() UNIQUE IDENTIFIER execute(org.apache.http.client.methods.HttpUriRequest) NETWORK be sent to a remote server per an HTTP POST request.  This malicious flow starts by reading SMS message, then writes to a BasicNameValuePair object, to org.apache.http.client.HttpClient: org.apache.http.client.HttpClient: org.apache.http.client.HttpClient: org.apache.http.client.methods.HttpUriRequest)  NETWORK be sent to a remote server per an HTTP POST request.  This malicious flow starts by reading SMS message, then writes to a BasicNameValuePair object, to org.apache.http.client.HttpClient: org.apache.http.client.HttpClient: org.apache.http.client.methods.HttpUriRequest)  NETWORK by the class smsReceiver runs.					1 - 1		
fakebank_andro id_samp 4 android.telephony.SmsMessage: java.lang.String getDisplayMessageBody() SMS MMS by reading SMS message, then writes to a BasicNameValuePair object, to org.apache.http.client.HttpClient: org.apache.http.lttpResponse execute(org.apache.http.client.methods.HttpUriRequest) NETWORK by the class smsReceiver runs.							This malicious flow starts by reading the serial number of the SIM card, then writes to a List. This list will
fakebank_andro   org.apache.http.client: org.apache.http.client: org.apache.http.client.HttpClient: org.apache.http.clien	id_samp	3	getSimSerialNumber()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	
id_samp							This malicious flow starts by reading SMS message, then writes to a BasicNameValuePair object, then a
	_						HashMap. The hashmap will be sent to a rmote server per an HTTP POST request when a Thread created
		4	android.telephony.SmsMessage: java.lang.String getDisplayMessageBody()	SMS MMS		NETWORK	by the class smsReceiver runs.
fakebank_andro com.example.service: lnstallService: void						l	
id_samp 5 java.io.File: void <init an="" apk="" file="" flow="" from="" installs="" intent="" malicous="" sdcard.<="" startactivity(android.content.intent)="" td="" this=""><td>Id_samp</td><td> 5</td><td>Java.io.File: void <init< td=""><td>FILE</td><td>startActivity(android.content.Intent)</td><td>INTENT</td><td>This malicous flow installs an apk from sdcard.</td></init<></td></init>	Id_samp	5	Java.io.File: void <init< td=""><td>FILE</td><td>startActivity(android.content.Intent)</td><td>INTENT</td><td>This malicous flow installs an apk from sdcard.</td></init<>	FILE	startActivity(android.content.Intent)	INTENT	This malicous flow installs an apk from sdcard.

		android.telephony.SmsMessage: android.telephony.SmsMessage		org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
fakedaum	1	createFromPdu(byte[])	SMS MMS	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads a received SMS and uploads it to an HTTP Server.
		android.telephony.TelephonyManager: java.lang.String		org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
fakedaum	2	getSimSerialNumber()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads the IMEI and uploads it to an HTTP Server.
		android.telephony.gsm.SmsMessage: android.telephony.gsm.SmsMessage				
fakemart	1	createFromPdu(byte[])	SMS MMS	java.net.URLConnection: java.io.InputStream getInputStream()	NETWORK	This malicious flow reads incoming SMSs and posts them to a URL.
		android.telephony.gsm.SmsMessage: android.telephony.gsm.SmsMessage				
fakemart	2	createFromPdu(byte[])	SMS MMS	java.net.URLConnection: java.io.InputStream getInputStream()	NETWORK	This malicious flow reads incoming SMSs and posts them to a URL.
		android.telephony.TelephonyManager: java.lang.String				
fakeplay	1	getNetworkOperatorName()	NETWORK INFORMATION	java.io.OutputStream: void write(byte[],int,int)	EMAIL	This malicious flow leaks telephoney network information via email.
fakeplay	2	android.telephony.TelephonyManager: java.lang.String getLine1Number()	UNIQUE IDENTIFIER	java.io.OutputStream: void write(byte[],int,int)	EMAIL	This malicious flow leaks telephone number via email.
такеріау	-	and outteephony, relephony wanager, java.lang. 5thing gettine twumber()	UNIQUE IDENTIFIER	org.apache.http.impl.client.DefaultHttpClient:	EIVIAIL	This mailclous now leaks telephone number via email.
				org.apache.http.HttpResponse		
faketaobao	1	android.widget.EditText: android.text.Editable getText()	ACCOUNT INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads user given taobao password and sends to a remote server.
таксаобао	-	and old. Widget. Edit rext. and old. text. Editable get rext()	ACCOUNT IN CHINATION	org.apache.http.impl.client.DefaultHttpClient:	IVETWORK	This mancious now reads user given tabbab password and serias to a remote server.
				org.apache.http.HttpResponse		
faketaobao	2	android.widget.EditText: android.text.Editable getText()	ACCOUNT INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads user given taobao username and sends to a remote server.
таксаобао	_	and old. Widget. Edit rext. and old. text. Editable get rext()	ACCOUNT IN CHINATION	org.apache.http.impl.client.DefaultHttpClient:	IVETWORK	This multious now reads user given tabbab discrimine and series to a remote server.
				org.apache.http.HttpResponse		
faketaobao	3	android.widget.EditText: android.text.Editable getText()	ACCOUNT INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads user's ID and sends it to a remote server.
ranctaobao		and out magetizate ext. and out textical case get ext()	TICCOUNT IN CHARACTER	org.apache.http.impl.client.DefaultHttpClient:	THE THE THE	This mandred now reads user 5 to and series to a remote server.
				org.apache.http.HttpResponse		
faketaobao	4	android.widget.EditText: android.text.Editable getText()	ACCOUNT INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flows reads user given password and sends to a remote server.
		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		org.apache.http.impl.client.DefaultHttpClient:		, and the second
				org.apache.http.HttpResponse		This malicious flow reads the phone number and puts it into a ArrayList. This list is sent to a remote
godwon_samp	1	android.telephony.TelephonyManager: java.lang.String getLine1Number()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	server via a HTTP Post request.
				org.apache.http.impl.client.DefaultHttpClient:		
				org.apache.http.HttpResponse		This malicious flow reads the device Id when the phone number is empty. This device Id is stored in a
godwon_samp	2	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	ArrayList. This list is sent to a remote server via a HTTP Post request.
				org.apache.http.impl.client.DefaultHttpClient:		This malicious flow exists in the onReceive callback method of a BroadcastReciver, it reads SMS
		android.content.Intent: java.io.Serializable		org.apache.http.HttpResponse		messages to array, appends all messages to a string, stores it into a List and sends the list to a remote
godwon_samp	3	getSerializableExtra(java.lang.String)	SMS MMS	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	server via a HTTP POST request.
				org.apache.http.impl.client.DefaultHttpClient:		This malicious flow exists in the onReceive callback method of a BroadcastReciver, it reads SMS
		android.content.Intent: java.io.Serializable		org.apache.http.HttpResponse		messages to array, writes the sender's address into a private field, stores it into a List and sends the list
godwon_samp	4	getSerializableExtra(java.lang.String)	SMS MMS	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	to a remote server via a HTTP POST request.
				org.apache.http.impl.client.DefaultHttpClient:		
				org.apache.http.HttpResponse		
godwon_samp	5	android.telephony.TelephonyManager: java.lang.String getLine1Number()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads the phone number and sends it to a rmote server via a HTTP Post request.
				org.apache.http.impl.client.DefaultHttpClient:		
				org.apache.http.HttpResponse		This malicious flow reads device id when the phone number is empty, and it send the device id to a
godwon_samp	6	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	remote server via a HTTP Post request.
				android.database.sqlite.SQLiteDatabase: long		This malicious ICC flow crosses one activity and two services. It starts by reading advertisement
hummingbad_a	1.			insert(java.lang.String,java.lang.String,android.content.ContentValues		information from data base into a JSONObject, when the user performs a click, new advertisements will
ndroid_samp	1	android.database.Cursor: java.lang.String getString(int)	DATABASE	]	DATABASE	be downloaded and saved into data base.
				android.database.sqlite.SQLiteDatabase: int		This malicious ICC flow crosses one activity and two services. It starts by reading advertisement
hummingbad_a				update(java.lang.String,android.content.ContentValues,java.lang.Strin		information from data base into a JSONObject, when the user performs a click, new advertisements will
ndroid_samp	2	android.database.Cursor: java.lang.String getString(int)	DATABASE	g,java.lang.String[])	DATABASE	be downloaded and updated in the data base.
ially same	1	and said talanham. Talanham Manager: Invalid State + David 199	LINIOUE IDENTIFIED	java.util.concurrent.ThreadPoolExecutor: java.util.concurrent.Future	NETWORK	The malicious flow aves the the phone's device ID in a public static field and leaks the device ID into an
jollyserv	1	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	submit(java.lang.Runnable)	NETWORK	HTTP request.
overlaulecke -2				andraid wobkit WohViow; void		This malicious flow reads the device id and uses it as the return value of the method imei() of the class
overlaylocker2_	1	andraid tolophony Tolophony Managary invalang String get Douisedd/	UNIQUE IDENTIFIER	android.webkit.WebView: void	CDITICAL ELINICTION	MeSetting. An object of MeSetting is injected into the WebView by addJavascriptInterface. The method
android_samp	1	android.telephony.TelephonyManager: java.lang.String getDeviceld()	UNIQUE IDENTIFIEK	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	imei() can be invoked via JavaScript code.  This malicious flow reads the device id and stores it as a field of an Telephonylafe object. This
						This malicious flow reads the device id and stores it as a field of an Telephonylnfo object. This
overlaylocker2				android.webkit.WebView: void		information will be used as the return value of the method dualsim() of the class MeSystem. An object o MeSystem is injected into the WebView by addJavascriptInterface. The method dualsim() can be
android_samp	2	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL ELINICTION	invoked via JavaScript code.
anui viu_Samp	4	and old telephony, relephonyiwanager: Java.lang.string getDeviCeId()	ONIQUE IDENTIFIER	aumavascriptinterracetjava.iang.object,java.iang.stringj	CRITICAL FUNCTION	invokeu via JavaScript Coue.
overlaylocker2		java.lang.Class: java.lang.reflect.Method		java.lang.reflect.Method: java.lang.Object		
android samp	3	getDeclaredMethod(java.lang.String,java.lang.Class[])	CRITICAL FUNCTION	invoke(java.lang.Object,java.lang.Object[])	CRITICAL FLINCTION	This malicious flow uses Java Reflection to execute code which gets the service for a selected sim card.
anaroia_samp	٦	Berneria carreta or flavariang. String, Javariang. Class[]]	CHITICAL FORCTION	invokeljavanang.object,javanang.object[])	CHITICALTONCHON	This mancious now ascassava herection to execute code winds gets the service for a selected sillicatu.

overlaylocker2		java.lang.Class: java.lang.reflect.Method		java.lang.reflect.Method: java.lang.Object		
android_samp	4	getDeclaredMethod(java.lang.String,java.lang.Class[])	CRITICAL FUNCTION	invoke(java.lang.Object,java.lang.Object[])	CRITICAL FUNCTION	This malicious flow uses Java Reflection to execute code which sends SMS messages.
overlaylocker2_	_	lana la Ellahan deCharana and distrib	511.5	is as in Data Outrout Standard and the shift	NETWORK	This selection of the search o
android_samp	5	java.io.FileInputStream: void <init< td=""><td>FILE</td><td>java.io.DataOutputStream: void flush()</td><td>NETWORK</td><td>This malicious flow reads file and sends the file to a remote server via an HTTP POST request.  This malicious flow reads file from a given path and and uses it as the return value of the method read()</td></init<>	FILE	java.io.DataOutputStream: void flush()	NETWORK	This malicious flow reads file and sends the file to a remote server via an HTTP POST request.  This malicious flow reads file from a given path and and uses it as the return value of the method read()
overlaylocker2				android.webkit.WebView: void		of the class MeFile. An object of MeFile is injected into the WebView by addJavascriptInterface. The
android samp	6	java.io.FileInputStream: void <init< td=""><td>FILE</td><td>addJavascriptInterface(java.lang.Object,java.lang.String)</td><td>CRITICAL FUNCTION</td><td>method read() can be invoked via JavaScript code.</td></init<>	FILE	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	method read() can be invoked via JavaScript code.
overlaylocker2_		android.telephony.SmsMessage: android.telephony.SmsMessage		android.content.Context: android.content.ComponentName		
android_samp	7	createFromPdu(byte[])	SMS MMS	startService(android.content.Intent)	INTENT	This malicious flow monitors incoming SMSs.
				and and contain Make Make and a		
overlaylocker2_ android_samp	8	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	android.webkit.WebView: void addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	Negative flow
anaroia_samp		and outceephony. recephony wanager. Java.lang. String get bevice a ()	ONIQUE IDENTIFIER	addiavaseriptimeerracetjava.iang.object.java.iang.string/	CRITICAL FORCTION	regarie now.
overlaylocker2				android.webkit.WebView: void		
android_samp	9	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	Negative flow.
overlaylocker2_				android.webkit.WebView: void		
android_samp	10	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	Negative flow.
overlaylocker2				android.webkit.WebView: void		
android samp	11	android.telephony.TelephonyManager: java.lang.String getDeviceld()	UNIQUE IDENTIFIER	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	Negative flow.
overlaylocker2_				android.webkit.WebView: void		
android_samp	12	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	Negative flow.
overlaylocker2_	4.2	and a side to be a second to the second to t	LINIOUE IDENTIFIED	android.webkit.WebView: void	CDITICAL FUNCTION	No service floor
android_samp	15	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	Negative now.
overlaylocker2_				android.webkit.WebView: void		
android_samp	14	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	Negative flow.
overlaylocker2_				android.webkit.WebView: void		
android_samp	15	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	Negative flow.
overlaylocker2				android.webkit.WebView: void		
android samp	16	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	Negavie flow.
overlaylocker2_				android.webkit.WebView: void		
android_samp	17	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	Negative flow.
overlaylocker2_	10	andraid content Intent: invalanc String gotStringEytraliava lang String	OTHER DATA	android.webkit.WebView: void	CRITICAL FUNCTION	Nogative flow
android_samp	10	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL PONCTION	negative now.
overlaylocker2_				android.webkit.WebView: void		
android_samp	19	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	Negative flow.
						This malicious partial ICC flow starts by reading credit card number from user input and writes to a field
						of the Activity. When the user performs a click on a button, the information stealing starts. The
overlay_android		and the second s	A CCOUNT INFORMATION	exts.whats.activities.Cards: android.content.ComponentName	INITENIT	information will be written to a JSONObject which will be stored into an Intent with the name data. With
_samp	1	exts.whats.activities.Cards: android.view.View findViewById(int)	ACCOUNT INFORMATION	startService(android.content.Intent) org.apache.http.impl.client.DefaultHttpClient:	INTENT	this intent a new SendService which leaks the card number will be started.
overlay android				org.apache.http.HttpResponse		This malicious partial ICC flow reads credit card information from the Intent with which the SendSevice is
	2	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	ACCOUNT INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	started and sends it to a remote server per an HTTP POST request.
		, 5 55 5 5 5 6 6				This malicious partial ICC flow starts by reading SMS messages from the Intent with which the
overlay_android				android.content.Context: android.content.ComponentName		MessageReceiver is started. The messages will be written to a Map which will be stored into an Intent.
_samp	3	android.content.Intent: android.os.Bundle getExtras()	SMS MMS	startService(android.content.Intent)	INTENT	With this intent a new SendService which leaks the card number will be started.
Lauratan C. 11				org.apache.http.limpl.client.DefaultHttpClient:		This will be a set of the second of the seco
overlay_android samp	4	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	SMS MMS	org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious partial ICC flow reads SMS messages from the Intent with which the SendService is started and sents it to a remote server per an HTTP Post request.
_301116	17	and outcome interior javanang. String getstring extra ajjavanang. String)	STATE INTERIO	excessed a graph contraction of the contraction of	- TET TO AIR	and sente to a remote server per unitin it operequest.

overlay_android	ıl			exts.whats.activities.Cards: android.content.ComponentName		
samp	5	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	startService(android.content.Intent)	INTENT	Negative flow.
overlay_android				exts.whats.activities.Cards: android.content.ComponentName		
	6	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	OTHER DATA	startService(android.content.Intent)	INTENT	Negative flow.
	1	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	java.io.DataOutputStream: void writeUTF(java.lang.String)	NETWORK	This malicious flow reads device id and sends to a remote server via socket.
	2	java.io.FileInputStream: void <init< td=""><td>FILE</td><td>java.io.DataOutputStream: void write(byte[],int,int)</td><td>NETWORK</td><td>This malicious flow reads JPG images from sd card and send to a remote server via socket</td></init<>	FILE	java.io.DataOutputStream: void write(byte[],int,int)	NETWORK	This malicious flow reads JPG images from sd card and send to a remote server via socket
	3	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	java.io.DataOutputStream: void write(byte[],int,int)	NETWORK	Negative flow.
,	4	java.io.FileInputStream: void <init< td=""><td>FILE</td><td>java.io.DataOutputStream: void write(byte[j,int,int)</td><td>NETWORK</td><td>Negative flow.</td></init<>	FILE	java.io.DataOutputStream: void write(byte[j,int,int)	NETWORK	Negative flow.
	5		FILE	java.io.DataOutputStream: void writeOTT (java.lang.String)		
	1	java.io.FileInputStream: void <init android.telephony.gsm.gsmcelllocation:="" getlac()<="" int="" td=""><td>LOCATION INFORMATION</td><td></td><td>NETWORK LOG</td><td>Negative flow. This flow logs location information.</td></init>	LOCATION INFORMATION		NETWORK LOG	Negative flow. This flow logs location information.
proxy_samp	1	android.telephony.gsm.GsmCellLocation: Int getLac()	LOCATION INFORMATION	android.util.Log: int i(java.lang.String,java.lang.String)	LUG	This now logs location information.
proxy_samp	2	android.telephony.gsm.GsmCellLocation: int getLac()	LOCATION INFORMATION	org.apache.http.client.HttpClient: org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads location information and writes its into a private field, then sends it to a remote server per an HTTP POST request.
				org.apache.http.impl.client.DefaultHttpClient: org.apache.http.HttpResponse		This malicious flow reads a log file from external storage line by line and sends the content to a remote server. Each line of the file is written to an ArrayList. This ArrayList is used to construct the HTTP entity
proxy_samp	3	java.io.File: void <init< td=""><td>FILE</td><td>execute(org.apache.http.client.methods.HttpUriRequest)</td><td>NETWORK</td><td>which will be sent to a remote server.</td></init<>	FILE	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	which will be sent to a remote server.
		android.accounts.AccountManager: android.accounts.Account[]				
proxy_samp	4	getAccounts()	ACCOUNT INFORMATION	java.io.BufferedWriter: void write(java.lang.String)	FILE	This malicious flow reads account information and writes it to a log file.
		android.accounts.AccountManager: android.accounts.Account[]				This malicious flow reads account information and writes gmail acount into a private field. The gmail
proxy_samp	5	getAccounts()	ACCOUNT INFORMATION	java.io.BufferedWriter: void write(java.lang.String)	FILE	acount name will be logged into a file.
		android.content.ContentResolver: android.database.Cursor query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String,java.lang.String[],jav				
proxy_samp	6	a.lang.String)	DATABASE	java.io.BufferedWriter: void write(java.lang.String)	FILE	This malicious flow reads sms from database and writes it a log file.
proxy_samp	7	android.net.wifi.WifiManager: android.net.wifi.WifiInfo getConnectionInfo()	NETWORK INFORMATION	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow reads wifi connection information and logs it.
p,p	Ė	g				
proxy_samp	8		SMS MMS	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow starts in a BroadcastReceiver. It reads sms message and logs it.
		android.accounts.AccountManager: android.accounts.Account[]				
proxy_samp	9	getAccounts()	ACCOUNT INFORMATION	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs account information on the device.
				org.apache.http.impl.client.DefaultHttpClient:		
		android.accounts.AccountManager: android.accounts.Account[]		org.apache.http.HttpResponse		
proxy_samp	10	getAccounts()	ACCOUNT INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow sends Google account name to a remote server via a HTTP POST request.
		Ī i				
proxy_samp	11	android.telephony.SmsMessage: java.lang.String getDisplayMessageBody()	SMS MMS	java.io.BufferedWriter: void write(java.lang.String)	FILE	This malicious flow reads incomming SMS message and writes to a log file.
nrow, camp	12	android.net.wifi.WifiManager: android.net.wifi.WifiInfo getConnectionInfo()	NETWORK INCORMATION	java.io.BufferedWriter: void write(java.lang.String)	FILE	This malicious flow reads WIFI information and writes to a log file.
proxy_samp	12	android.content.ContentResolver: android.database.Cursor	NETWORK INTORNATION	java.io.buriereuwriter. void writetjava.iang.string/	TILL	This maintious now reads with mormation and writes to a log me.
		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				
proxy_samp	13	a.lang.String)	DATABASE	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs incoming SMS messages.
		android.content.ContentResolver: android.database.Cursor				
		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				This malicious flow saves phone call logs to a log file when the ProxyService is destoryed. This file will be
proxy_samp	14	a.lang.String)	DATABASE	java.io.BufferedWriter: void write(java.lang.String)	FILE	later uploaded to a remote server
		android.content.ContentResolver: android.database.Cursor				
		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				This malicious flow saves phone call logs to a log file when the ProxyService is destroyed. This file will be
proxy_samp	15	a.lang.String)	DATABASE	java.io.BufferedWriter: void write(java.lang.String)	FILE	later uploaded to a remote server.
	16	java.io.File: void <init< td=""><td>FILE</td><td>java.io.BufferedWriter: void write(java.lang.String)</td><td>FILE</td><td>Negative flow.</td></init<>	FILE	java.io.BufferedWriter: void write(java.lang.String)	FILE	Negative flow.
		android.content.ContentResolver: android.database.Cursor				
		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				
proxy_samp	17	a.lang.String)	DATABASE	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs phone calls.
	-	android.content.ContentResolver: android.database.Cursor				
		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				
proxy_samp	18	a.lang.String)	DATABASE	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	This malicious flow logs phone calls.
	10	andraid act wife Wife Managers and said act wife Wife Info and a control of the	NETWORK INCORNATION	andraid util Lagrint if in a lang Chrise in a lang Chris	100	Nametics flow
. /= .	19	android.net.wifi.WifiManager: android.net.wifi.WifiInfo getConnectionInfo()		android.util.Log: int i(java.lang.String,java.lang.String)	LOG	Negative flow.
	20	java.io.File: void <init< td=""><td>FILE</td><td>android.util.Log: int i(java.lang.String,java.lang.String)</td><td>LOG</td><td>Negative flow.</td></init<>	FILE	android.util.Log: int i(java.lang.String,java.lang.String)	LOG	Negative flow.
remote_control		android.location.LocationManager: android.location.Location				This malicious flow reads location information (longitude) and saves it a text file which will be later
	1	getLastKnownLocation(java.lang.String)	LOCATION INFORMATION	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	uploaded to a remote server.
remote_control		android.location.LocationManager: android.location.Location				This malicious flow reads location information (latitude) and saves it a text file which will be later
_	10	getLastKnownLocation(java.lang.String)	LOCATION INFORMATION	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	uploaded to a remote server.
	2	gettastinown zotation (javanan g. st. m.g)		Jeremen Bremen,	1100	•
_smack remote_control	_	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	This malicious flow reads SMS information (address) and saves it a text file which will be later uploaded to a remote server.

remote control						This malicious flow reads SMS information (message body) and saves it a text file which will be later
smack	4	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	uploaded to a remote server.
remote_control		, , , , , , , , , , , , , , , , , , , ,		,, , , , , , , , , , , , , , , , , , , ,		This malicious flow reads SMS information (message type) and saves it a text file which will be later
	5	android.database.Cursor: int getInt(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	uploaded to a remote server.
remote_control						This malicious flow reads SMS information (date) and saves it a text file which will be later uploaded to a
_smack	6	android.database.Cursor: long getLong(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	remote server.
remote_control						This malicious partial flow reads call record (number) and saves it a text file which will be later uploaded
_smack	7	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	to a remote server.
remote_control						This malicious partial flow reads call record (name) and saves it a text file which will be later uploaded to
_smack	8	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	a remote server.
remote_control						This malicious flow reads call record (type) and saves it a text file which will be later uploaded to a
_smack	9	android.database.Cursor: int getInt(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	remote server.
remote_control						This malicious flow reads call record (date) and saves it a text file which will be later uploaded to a
_smack	10	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	remote server.
remote_control						This malicious flow reads contact information and saves it a text file which will be later uploaded to a
_smack	11	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	remote server.
remote_control						This malicious flow reads contact information and saves it a text file which will be later uploaded to a
_smack	12	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	remote server.
remote_control						This malicious flow reads calender information and saves it a text file which will be later uploaded to a
smack	13	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	remote server.
remote control						This malicious flow reads calender information and saves it a text file which will be later uploaded to a
	14	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	remote server.
remote_control		, , ,		, , , , ,		This malicious flow reads calender information and saves it a text file which will be later uploaded to a
	15	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	remote server.
remote control		, , ,				This malicious flow reads calender information and saves it a text file which will be later uploaded to a
_	16	android.database.Cursor: long getLong(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	remote server.
remote_control		5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5		, ,		This malicious flow reads calender information and saves it a text file which will be later uploaded to a
	17	android.database.Cursor: java.lang.String getString(int)	DATABASE	java.io.FileWriter: java.io.Writer append(java.lang.CharSequence)	FILE	remote server.
		,		, ,		
repane	1	java.io.File: void <init< td=""><td>FILE</td><td>android.content.Context: void startActivity(android.content.Intent)</td><td>INTENT</td><td>This malicious flows installs an APK that is shipped with the App.</td></init<>	FILE	android.content.Context: void startActivity(android.content.Intent)	INTENT	This malicious flows installs an APK that is shipped with the App.
		android.content.ContentResolver: android.database.Cursor		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		Provide the second seco
		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				
roidsec	1	a.lang.String)	DATABASE	java.io.OutputStream: void write(byte[])	NETWORK	This malicious flow reads phone numbers from call logs and sends to a remote server via socket.
		android.content.ContentResolver: android.database.Cursor		, and the second		
		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				
roidsec	2	a.lang.String)	DATABASE	java.io.OutputStream: void write(byte[])	NETWORK	This malicious flow reads phone numbers from contact and sends to aremote server via socket.
roidsec	-		5717157152	javanorousputstreum rotu mite(s) tett)	I TET WORK	This managed now reads prome numbers from contact and series to dremote server his society.
roidsec	3	android.content.pm.PackageManager: java.util.List getInstalledPackages(int)	SYSTEM SETTINGS	java.io.OutputStream: void write(byte[])	NETWORK	This malicious flow reads installed package name and sends to a remote server via socket.
roidsec		and outcontent print dendger and agent jurisdatiness get instances dendges (inty	51512111521111165	javanorousputstreum rotu mite(s) tett)	I TET WORK	This managed now reads instance passage name and series to a remote server via source.
roidsec	4	android.content.pm.PackageManager: java.util.List getInstalledPackages(int)	SYSTEM SETTINGS	java.io.OutputStream: void write(byte[])	NETWORK	This malicious flow reads information about installed apps and sends it to a remote server.
roidsec		android.content.ContentResolver: android.database.Cursor	51512111521111105	jarano.outpubiteum roid imte(o)te(j)	11211101111	This manages now reads information about instance apps and series a territore server.
		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				
roidsec	5	a.lang.String)	DATABASE	java.io.OutputStream: void write(byte[])	NETWORK	This malicious flow reads incoming SMSs and sends them to a remote server.
Tolusee	-	android.content.ContentResolver: android.database.Cursor	DATABASE	java.io.outputstream. void write(byte[j)	IVETVOIN	This mancious now redus incoming swiss and serius them to a remote server.
		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				
roidsec	6	a.lang.String)	DATABASE	java.io.OutputStream: void write(byte[])	NETWORK	This malicious flow reads outgoing SMSs and sends them to a remote server.
. Slusec	9		J ADAJE	jarano.outputotreum. void write(pyte(j)		This manages now reads outgoing smost and series them to a remote server.
				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
samsapo	1	android.database.Cursor: java.lang.String getString(int)	DATABASE	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads SMS address and sends it to a remote server via HTTP Post request.
Janijapu	1	android.telephony.gsm.SmsMessage: android.telephony.gsm.SmsMessage	DATABASE	com.android.tools.system.MyPostRequest: android.os.AsyncTask	INC I WOULD	This mancious now reads sivis address and serius it to difemote server via mitr rost request.
camcano	2	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	SMS MMS	1	NETWORK	This malicious flow reads incoming SMSs and conds them to a remote conversio LTTD request
samsapo	2	createFromPdu(byte[])	CIVIIVI CIVIC	execute(java.lang.Object[])	INETWUKK	This malicious flow reads incoming SMSs and sends them to a remote server via HTTP request.
samsano	2	android.content.Context: java.lang.Object getSystemService(java.lang.String)	SVSTEM SETTINGS	java.lang.reflect.Method: java.lang.Object invoke(java.lang.Object,java.lang.Object[])	CRITICAL FUNCTION	This malicious flow uses reflection to terminate calls from user's contact.
samsapo	3	and out content. context: java.lang.object getsystemservice(java.lang.string)	STSTEIN SETTINGS	invoketjava.iang.Object,java.iang.Object[])	CRITICAL FUNCTION	This mancious now uses reflection to terminate cans from user's contact.
		andraid talanhany sam Cmahlassasa, d1d t-1d		ara anasha http aliant Http://iont.		
		android.telephony.gsm.SmsMessage: android.telephony.gsm.SmsMessage	CNAC NANAC	org.apache.http.client.HttpClient: org.apache.http.HttpResponse	NETWORK	This melicious flavorands incoming CMC massage to a variety service a LITTE SOCT
samsapo	4	createFromPdu(byte[])	SMS MMS	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow sends incoming SMS message to a remote server via a HTTP POST.
	_	and said content Contouts in a lang Object to the contout in a lang Object to th	CVCTEM CETTINGS	java.lang.reflect.Method: java.lang.Object	CDITICAL FUNCTION	Negative flavo
samsapo	5	android.content.Context: java.lang.Object getSystemService(java.lang.String)	SYSTEM SETTINGS	invoke(java.lang.Object,java.lang.Object[])	CRITICAL FUNCTION	Negative flow.
				and and an artist Contain Welling and 11		This reliable of the second of
		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		android.content.ContentValues: void		This malicious flow reads user name from the input field and stores into database, which will be queried
save_me	1	android.widget.EditText: android.text.Editable getText()	ACCOUNT INFORMATION	put(java.lang.String,java.lang.String)	DATABASE	by the service called CO running in the background and sent to a remote server via HTTP requests.

				- deid - start Costanticles		This malicious flow reads phone number from the input field and stores into database, which will be
save_me	2	android.widget.EditText: android.text.Editable getText()	ACCOUNT INFORMATION	android.content.ContentValues: void put(java.lang.String,java.lang.String)	DATABASE	queried by the service called CO running in the background and sent to a remote server via HTTP requests.
save_me		android.database.sqlite.SQLiteDatabase: android.database.Cursor	ACCOUNT INFORMATION	android.telephony.SmsManager: void	DATABASE	requests.
		query(java.lang.String,java.lang.String[],java.lang.String,java.lang.String[],jav		sendTextMessage(java.lang.String,java.lang.String,java.lang.String,and		This malicious flow reads the phone number from database and sends spam sms message to this phone
save_me	3	a.lang.String,java.lang.String)	DATABASE	roid.app.PendingIntent,android.app.PendingIntent)	SMS MMS	number.
		android.database.sqlite.SQLiteDatabase: android.database.Cursor				
save_me	4	query(java.lang.String,java.lang.String[],java.lang.String,java.lang.String[],java.lang.String,java.lang.String,java.lang.String,java.lang.String,java.lang.String	DATABASE	org.apache.http.client.HttpClient: org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads user name from database and puts into a ArrayList of name value pairs. This list is sent to a remote server per an HTTP Post request.
		android.database.sqlite.SQLiteDatabase: android.database.Cursor				
save_me	5	query(java.lang.String,java.lang.String[],java.lang.String,java.lang.String[],java.lang.String,java.lang.String,java.lang.String)	DATABASE	org.apache.http.client.HttpClient: org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads phone number from database and puts into a ArrayList of name value pairs.  This list is sent to a remote server per an HTTP Post request.
save_me	6	android.net.wifi.WifiInfo: java.lang.String getMacAddress()	NETWORK INFORMATION	org.apache.http.client.HttpClient: org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads MAC address and writes it into a public field of the service CO. Value from this field is put into a ArrayList of name value pairs. This list is sent to a remote server per an HTTP Post request.
				org.apache.http.client.HttpClient: org.apache.http.HttpResponse	NETHON	This malicious flow reads country information and writes it into a public field of the service CO. Value from this field is put into a ArrayList of name value pairs. This list is sent to a remote server per an HTTP
save_me	7	android.telephony.TelephonyManager: java.lang.String getSimCountryIso()	LOCATION INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	Post request.
		android.database.sqlite.SQLiteDatabase: android.database.Cursor query(java.lang.String,java.lang.String[],java.lang.String,java.lang.String[],jav		org.apache.http.client.HttpClient: org.apache.http.HttpResponse		This malicious flow reads user name from database and puts into a ArrayList of name value pairs. This list
save me	8	a.lang.String,java.lang.String)	DATABASE	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	is sent to a remote server per an HTTP Post request.
	1	android.database.sqlite.SQLiteDatabase: android.database.Cursor		(- 8p		
		query(java.lang.String,java.lang.String[],java.lang.String,java.lang.String[],jav		org.apache.http.client.HttpClient: org.apache.http.HttpResponse		This malicious flow reads phone number from database and puts into a ArrayList of name value pairs.
save_me	9	a.lang.String,java.lang.String,java.lang.String)	DATABASE	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This list is sent to a remote server per an HTTP Post request.
		android.database.sqlite.SQLiteDatabase: android.database.Cursor				
		query(java.lang.String,java.lang.String[],java.lang.String,java.lang.String[],jav		org.apache.http.client.HttpClient: org.apache.http.HttpResponse		This malicious flow reads MAC address from database and puts into a ArrayList of name value pairs. This
save_me	10	a.lang.String,java.lang.String,java.lang.String)	DATABASE	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	list is sent to a remote server per an HTTP Post request.
				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		This malicious ICC flow starts by reading phone number from the input field and stores the value into an intent. With this intent a new activity is stareted. Phone number is read from the intent, written into an
save me	11	android.widget.EditText: android.text.Editable getText()	ACCOUNT INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	ArrayList which is sent to a remote server via an HTTP post request.
Save_me		und out widget. Edit Text. und out text. Editable get Text()	ACCOUNT IN CHIMATION	execute(org.apacite.intep-citerit.inethous.intep-orinequest)	IVETVOIN	ArrayEst which is serie to a remote server via air iii ii post request.
save_me	12	android.widget.EditText: android.text.Editable getText()	ACCOUNT INFORMATION	org.apache.http.client.HttpClient: org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow sends user input to a remote server.
save_me	13	android.net.wifi.WifiInfo: java.lang.String getMacAddress()	NETWORK INFORMATION	org.apache.http.client.HttpClient: org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads MAC address and sends to a remote server.
				The state of the s		
				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
save_me	14	android.net.wifi.WifiInfo: java.lang.String getMacAddress()	NETWORK INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads Mac address and sends it to a remote server.
save_me	15	android.net.wifi.WifiInfo: java.lang.String getMacAddress()	NETWORK INFORMATION	org.apache.http.client.HttpClient: org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads MAC address and sends it to a remote server.
Save_me	15	android.database.sqlite.SQLiteDatabase: android.database.Cursor	NETWORK IN ORWATION	execute(org.apache.intp.chem.inethous.inteporthequest)	IVETVOIN	This manerous now reads wine address and series it to a remote server.
save_me	16	query(java.lang.String,java.lang.String[],java.lang.String,java.lang.String[],java.lang.String,java.lang.String]	DATABASE	org.apache.http.client.HttpClient: org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads user data from data base and sends it to a remote server.
		android.database.sqlite.SQLiteDatabase: android.database.Cursor				
		query(java.lang.String,java.lang.String[],java.lang.String,java.lang.String[],jav		org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
save_me	17	a.lang.String,java.lang.String,java.lang.String)	DATABASE	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads user data from database and sends it a remote server.
save_me	18	android.telephony.TelephonyManager: java.lang.String getSimCountryIso()	LOCATION INFORMATION	org.apache.http.client.HttpClient: org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow send country info to a remote server.
				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
save_me	19	android.telephony.TelephonyManager: java.lang.String getSimCountryIso()	LOCATION INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	Negative flow.
	20	and solid to lond boars. To londs on Managars in solang Chrising anti-in-County (sol)	LOCATION INFORMATION	org.apache.http.client.HttpClient: org.apache.http.HttpResponse	NETWORK	Newtine flow
save_me	20	android.telephony.TelephonyManager: java.lang.String getSimCountryIso()	LOCATION INFORMATION	execute(org.apacife.fittp.cliefit.fifetflous.fittpoffkequest)	NETWORK	Negative flow.
save_me	21	android.net.wifi.WifiInfo: java.lang.String getMacAddress()	NETWORK INFORMATION	org.apache.http.client.HttpClient: org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow sends MAC address to a remote server.

				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
save me	22	android.net.wifi.WifiInfo: java.lang.String getMacAddress()	NETWORK INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads MAC address and sends it to a remote server.
_		, , , , , , , , , , , , , , , , , , , ,				
				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
save_me	23	android.net.wifi.WifiInfo: java.lang.String getMacAddress()	NETWORK INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads MAC address and sends it to a remote server.
				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
save_me	24	android.net.wifi.WifiInfo: java.lang.String getMacAddress()	NETWORK INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads MAC address and sends it to a remote server.
				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
save_me	25	android.net.wifi.WifiInfo: java.lang.String getMacAddress()	NETWORK INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads MAC address and sends it to a remote server.
		android.database.sqlite.SQLiteDatabase: android.database.Cursor				
		query(java.lang.String,java.lang.String[],java.lang.String,java.lang.String[],jav		android.content.ContentValues: void		
save_me	26	a.lang.String,java.lang.String,java.lang.String)	DATABASE	put(java.lang.String,java.lang.String)	DATABASE	Negative flow.
		android.database.sqlite.SQLiteDatabase: android.database.Cursor				
		query(java.lang.String,java.lang.String[],java.lang.String,java.lang.String[],jav		android.content.ContentValues: void		
save_me	27	a.lang.String,java.lang.String)	DATABASE	put(java.lang.String,java.lang.String)	DATABASE	Negative flow.
		android.database.sqlite.SQLiteDatabase: android.database.Cursor		andraid content Content (alugarus 14		
	20	query(java.lang.String,java.lang.String[],java.lang.String,java.lang.String[],jav	DATABASE	android.content.ContentValues: void	DATABASE	Nagative flavo
save_me	28	a.lang.String,java.lang.String,java.lang.String)	DATABASE	put(java.lang.String,java.lang.String)	DATABASE	Negative flow.
				android.telephony.SmsManager: void sendTextMessage(java.lang.String,java.lang.String,java.lang.String,and		
save me	29	android.net.wifi.WifiInfo: java.lang.String getMacAddress()	NETWORK INFORMATION	roid.app.PendingIntent,android.app.PendingIntent)	SMS MMS	Negative flow.
Save_me		and old net will will not juvaliang. String get vide radii ess()	NETWORK IN ORMATION	rota.app.i chaingment,anarota.app.i chaingment/	SIVIS IVIIVIS	This malicious flow is a part of ICC flow which leaks contact information. The varaible var2 holds the
save me	30	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	com.savemebeta.Scan: void startActivity(android.content.Intent)	INTENT	contact name (see AddFriend class in com.savemebeta.Analyse.java).
	-			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		This malicious flow is a part of ICC flow which leaks contact information. The varaible var2 holds the
save me	31	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	CONTACT INFORMATION	com.savemebeta.Scan: void startActivity(android.content.Intent)	INTENT	contact name (see AddFriend class in com.savemebeta.Analyse.java).
_		, , , , , , , , , , , , , , , , , , , ,		i i		
						This malicious flow reads telephone number, stores it a static field and starts a service which listens to
scipiex	1	android.telephony.TelephonyManager: java.lang.String getLine1Number()	UNIQUE IDENTIFIER	java.io.OutputStream: void write(byte[])	NETWORK	the incoming SMSs. The telephone number will be sent to a remote server when SMSs are received.
		android.telephony.SmsMessage: android.telephony.SmsMessage				
scipiex	2	createFromPdu(byte[])	SMS MMS	java.io.OutputStream: void write(byte[])	NETWORK	This malicious flow reads received SMSs and send them to a remote server via a HTTP POST request.
		android.content.ContentResolver: android.database.Cursor				
		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				
scipiex	3	a.lang.String)	DATABASE	java.io.PrintWriter: void println(java.lang.String)	FILE	This malicious flow reads contacts and saves them to a file wich will be later sent to a remote server.
slocker_android				org.apache.http.client.HttpClient: org.apache.http.HttpResponse	l	This malicious flow reads device id and puts into a ArrayList of name value pairs. This list is sent to a
_samp	1	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	remote server per an HTTP Post request.
ala aluan an aluadal				and a second a latter of the City of the C		This walls to the same decise in and the late a Associate of a second control of the late and the same and th
slocker_android samp	2	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	org.apache.http.client.HttpClient: org.apache.http.HttpResponse execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads device id and puts into a ArrayList of name value pairs. This list is sent to a remote server per an HTTP Post request.
_samp	_	and outteephony, relephonywanager, java.lang.string getDeviceid()	UNIQUE IDENTIFIER	execute(org.apacite.itttp.client.ittetilous.itttportkequest)	INETWORK	Temote server per an inter rost request.
slocker_android				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
	3	android.widget.EditText: android.text.Editable getText()	ACCOUNT INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flows reads the inputed code and sends it to a remote server via HTTP POST request.
	-					
slocker android				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
_	4	android.widget.EditText: android.text.Editable getText()	ACCOUNT INFORMATION	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads the inputed code and sends to a remote server via HTTP POST request.
slocker_android				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		
_samp	5	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow reads device id and sends it to a remote server via an HTTP post.
smssend_packa				java.net.HttpURLConnection: java.io.OutputStream		
geInstaller	1	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	getOutputStream()	NETWORK	This malicious flow leaks the IMEI via an HTTP POST Request.
				android.telephony.SmsManager: void		
smssend_packa		android.telephony.SmsMessage: android.telephony.SmsMessage		sendTextMessage(java.lang.String,java.lang.String,java.lang.String,and		
geInstaller	2	createFromPdu(byte[])	SMS MMS	roid.app.PendingIntent,android.app.PendingIntent)	SMS MMS	This is is an implicit flow. When a certain SMS is received it sends another SMS.
				android.telephony.SmsManager: void		This model to the country of the cou
smssend_packa		and the latest and th	LINIOLIE IDENTIFIES	sendTextMessage(java.lang.String,java.lang.String,java.lang.String,and	CAAC AAAAC	This malicious flow sends device id in a SMS message back to the sender when a certian SMS message is
geInstaller	3	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	roid.app.PendingIntent,android.app.PendingIntent)	SMS MMS	received.
emecond most-				android.telephony.SmsManager: void		
smssend_packa geInstaller	4	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	sendTextMessage(java.lang.String,java.lang.String,java.lang.String,and roid.app.PendingIntent,android.app.PendingIntent)	SMS MMS	This malicious flow sends a SMS contains the device id.
Benistanei	-	and old telephony. Telephony ividinager. Java.lang.3tring getDeviceId()	ON QUE IDENTIFIER	Total application and total application application	21412 IAIIA12	This multilods now sends a sivis contains the device id.

			1	I		
smssend_packa	5	and a side to look and Tolonk and Management in the Chairman AD a size Id ()	LINIOUE IDENTIFIED	java.net.HttpURLConnection: java.io.OutputStream	NETWORK	This well-to a flavor and a decise to be a second
8	5	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	getOutputStream()	NETWORK	This malicious flow sends device Id to a remote server.
smssilience_fak	_		LINIO DE LE CONTRETE DE		NET WORK	
e_vertu		android.telephony.TelephonyManager: java.lang.String getLine1Number()	UNIQUE IDENTIFIER	java.io.PrintWriter: void write(java.lang.String)	NETWORK	This malicious flow leaks the users phone number to a remote server.
smssilience_fak		android.telephony.SmsMessage: android.telephony.SmsMessage	Chac hanac	in a la Deina Maria a consideration de la Carina N	NETWORK	This could be a second of the control of the contro
	2	createFromPdu(byte[])	SMS MMS	java.io.PrintWriter: void write(java.lang.String)	NETWORK	This malicous flow sends the contents of the retrieved sms to a remote server.
smssilience_fak	_		LINIO DE LE CONTRETE DE		NET WORK	N. C. B.
	3	android.telephony.TelephonyManager: java.lang.String getLine1Number()	UNIQUE IDENTIFIER	java.io.PrintWriter: void write(java.lang.String)	NETWORK	Negative flow.
smssilience_fak			LINIO DE LE CONTRETE DE		NET WORK	N. C. B.
e_vertu	4	android.telephony.TelephonyManager: java.lang.String getLine1Number()	UNIQUE IDENTIFIER	java.io.PrintWriter: void write(java.lang.String)	NETWORK	Negative flow.
		and a side and a set of Court and Donale and a side database Court				
smsstealer_kysn		android.content.ContentResolver: android.database.Cursor		and an about the allies the allies of the about the about the Barrers		This collision flowers dark house of any second and the second and
_assassincreed_		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav	DATABASE	org.apache.http.client.HttpClient: org.apache.http.HttpResponse	NETWORK	This malicious flow reads the types of sms messages and send to a remote server via an HTTP POST
android_samp	1	a.lang.String)	DATABASE	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	request.
		and a side and a set of Court and Donale and a side database Court				
smsstealer_kysn		android.content.ContentResolver: android.database.Cursor		ara anasha http slight little Client, ara anasha http little Dasnansa		This malicious flour roads the addresses of one massages and and to a romate convex via an UTTD DOST
_assassincreed_		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav	DATABACE	org.apache.http.client.HttpClient: org.apache.http.HttpResponse	NETWORK	This malicious flow reads the addresses of sms messages and send to a remote server via an HTTP POST
android_samp	2	a.lang.String)	DATABASE	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	request.
smsstealer_kysn						The last of the la
_assassincreed_	_			org.apache.http.client.HttpClient: org.apache.http.HttpResponse	NET WORK	This malicious flow reads the IMSI of the device and sends it to a remote server via an HTTP POST
android_samp	3	android.telephony.TelephonyManager: java.lang.String getSubscriberId()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	request.
smsstealer_kysn						
_assassincreed_			INITEDNIET COURCE	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	071150 0700 4 05	This malicious flow reads command and decrypts information from a remote server and store data into
android_samp	4	java.net.HttpURLConnection: java.io.InputStream getInputStream()	INTERNET SOURCE	android.content.SharedPreferences\$Editor: boolean commit()	OTHER STORAGE	shared perference.
smsstealer_kysn				android.telephony.gsm.SmsManager: void		
_assassincreed_	_		l	sendMultipartTextMessage(java.lang.String,java.lang.String,java.util.A		This malicious flow reads command and decrypts information from a remote server and sends a multi-
android_samp	5	java.net.HttpURLConnection: java.io.InputStream getInputStream()	INTERNET SOURCE	rrayList,java.util.ArrayList,java.util.ArrayList)	SMS MMS	part text based SMS.
				org.apache.http.client.HttpClient: org.apache.http.HttpResponse		This malicious flow reads the device id and puts it into a JSON object which will be sent to a remote
sms_google	1	com.google.elements.Utils: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	sever via HTTP request.
						The last 6 and 1 a
	_			org.apache.http.client.HttpClient: org.apache.http.HttpResponse		This malicious flow reads the device id and puts it into a JSON object which will be sent to a remote
sms_google	2	com.google.elements.Utils: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	sever via HTTP request.
				android.telephony.SmsManager: void		
	_			sendTextMessage(java.lang.String,java.lang.String,java.lang.String,and		
sms_google	3	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	SMS MMS	roid.app.PendingIntent,android.app.PendingIntent)	SMS MMS	This malicious partial flow listens to C&C command and sends SMS message in the background.
				android.telephony.SmsManager: void		
				sendTextMessage(java.lang.String,java.lang.String,java.lang.String,and		
sms_google	4	android.content.Intent: java.lang.String getStringExtra(java.lang.String)	SMS MMS	roid.app.PendingIntent,android.app.PendingIntent)	SMS MMS	This malicious partical flow listens to C&C command and sends SMS message.
				android.telephony.SmsManager: void		This well-time flow with in the collection will be the city of the collection of the
sms_send_locke				sendTextMessage(java.lang.String,java.lang.String,java.lang.String,and		This malicious flow exists in the onReceive callback method of a BroadcastReciver, it starts from reading
r_qqmagic	1	android.telephony.SmsMessage: java.lang.String getDisplayMessageBody()	SMS MMS	roid.app.PendingIntent,android.app.PendingIntent)	SMS MMS	a SMS message to a StringBuilder and ends by sending it via a text message to a strange number.
1				android.telephony.SmsManager: void		This malicious flow exists in the onReceive callback method of a BroadcastReciver, it starts from reading
sms_send_locke		android.telephony.SmsMessage: java.lang.String		sendTextMessage(java.lang.String,java.lang.String,java.lang.String,and	L <u>.</u>	the SMS sender's address (phone number or email) and ends by sending it via a text message to a
r_qqmagic	2	getDisplayOriginatingAddress()	CONTACT INFORMATION	roid.app.PendingIntent,android.app.PendingIntent)	SMS MMS	strange number.
						This malicious ICC flow cross one BroadcastReciver and one Service. This is only the partical flow in the
sms_send_locke				android.content.Context: android.content.ComponentName	l <u></u>	BroadcastReciver. The flow starts in a broadcast receiver by reading SMS message to a StringBuilder,
		android.telephony.SmsMessage: java.lang.String getDisplayMessageBody()	SMS MMS	startService(android.content.Intent)	INTENT	then uses this StringBuilder to construct an intent for a unknown service.
sms_send_locke				android.content.Context: android.content.ComponentName	l <u></u>	
qqagic		android.telephony.SmsMessage: java.lang.String getDisplayMessageBody()	SMS MMS	startService(android.content.Intent)	INTENT	Negative flow.
sms_send_locke				android.content.Context: android.content.ComponentName		
r_qqmagic	5	android.telephony.SmsMessage: java.lang.String getDisplayMessageBody()	SMS MMS	startService(android.content.Intent)	INTENT	
						This malicious ICC flow cross one BroadcastReciver and one Service. This is only the partical flow in the
sms_send_locke				android.content.Context: android.content.ComponentName		BroadcastReciver. The flow starts in a broadcast receiver by reading SMS message to a StringBuilder,
r_qqmagic	6	android.telephony.SmsMessage: java.lang.String getDisplayMessageBody()	SMS MMS	startService(android.content.Intent)	INTENT	then uses this StringBuilder to construct an intent for a unknown service.
						This malicious ICC flow cross one BroadcastReciver and one Service. This is only the partical flow in the
sms_send_locke				android.content.Context: android.content.ComponentName		BroadcastReciver. The flow starts in a broadcast receiver by reading SMS message to a StringBuilder,
r_qqmagic	7	android.telephony.SmsMessage: java.lang.String getDisplayMessageBody()	SMS MMS	startService(android.content.Intent)	INTENT	then uses this StringBuilder to construct an intent for a unknown service.

sms_send_locke				android.content.Context: android.content.ComponentName		This malicious ICC flow cross one BroadcastReciver and one Service. This is only the partical flow in the BroadcastReciver. The flow starts in a broadcast receiver by reading SMS message to a StringBuilder,
r_qqmagic	8	android.telephony.SmsMessage: java.lang.String getDisplayMessageBody()	SMS MMS	startService(android.content.Intent)	INTENT	then uses this StringBuilder to construct an intent for a unknown service.
stels_flashplaye		android.content.ContentResolver: android.database.Cursor				
r_android_upda		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav				
te	1	a.lang.String)	DATABASE	java.io.DataOutputStream: void write(byte[])	NETWORK	This malicious flow reads contacts and send them to a remote server via HTTP Post request.
stels_flashplaye						
r_android_upda te	2	android.telephony.TelephonyManager: java.lang.String getSubscriberId()	UNIQUE IDENTIFIER	java.io.DataOutputStream: void write(byte[])	NETWORK	This malicious flow reads device's IMSI number and stores it to the public field of class Settings. The IMSI number is sent a remote server via a HTTP POST request.
stels_flashplaye						
r_android_upda						
te	3	android.content.pm.PackageManager: java.util.List getInstalledPackages(int)	SYSTEM SETTINGS	java.io.DataOutputStream: void write(byte[])	NETWORK	This malicious leaks information about installed apks on the victim's device to a remote server.
						This malicious flow reads the IMEI and writs to a static field. IMEI storeds in this static field is sent it to a
tetus	1	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	java.net.HttpURLConnection: void connect()	NETWORK	remote server.
tetus	2	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	java.net.HttpURLConnection: void connect()	NETWORK	This malicious flow reads the IMEI and sends it to a server.
the interview				com.movieshow.down.Badaccents: void		
movieshow	1	java.net.URL: java.io.InputStream openStream()	FILE	startActivity(android.content.Intent)	INTENT	This malicious flow loads an apk file to external storage and installs this apk.
		android.content.ContentResolver: android.database.Cursor				
threatjapan ura		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav		java.io.PrintWriter: java.io.PrintWriter		
cto	1	a.lang.String)	DATABASE	append(java.lang.CharSequence)	FILE	This partial flow writes mail data into a file.
				org.apache.http.impl.client.DefaultHttpClient:		
threatjapan ura				org.apache.http.HttpResponse		
–	2	java.io.File: void <init< td=""><td>FILE</td><td>execute(org.apache.http.client.methods.HttpUriRequest)</td><td>NETWORK</td><td>This malicious flow sends a log file storing mail data to a remote server.</td></init<>	FILE	execute(org.apache.http.client.methods.HttpUriRequest)	NETWORK	This malicious flow sends a log file storing mail data to a remote server.
				The state of the s		
				org.springframework.web.client.RestTemplate:		
				org.springframework.http.ResponseEntity		This malicious flow starts by getting the external storage path and uploads data from the Viber
vibleaker andro				exchange(java.lang.String,org.springframework.http.HttpMethod,org.		directories to a remote server per an HTTP POST request. There are checks along the flow for checking if
id samp	1	android.os.Environment: java.io.File getExternalStorageDirectory()	FILE	springframework.http.HttpEntity,java.lang.Class,java.lang.Object[])	NETWORK	the app Viber is installed.
		, , , , , , , , , , , , , , , , , , , ,		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		
				org.springframework.web.client.RestTemplate:		
				org.springframework.http.ResponseEntity		
vibleaker andro				exchange(java.lang.String,org.springframework.http.HttpMethod,org.		This malicious flow reading videos from the Viber directories to a remote server per an HTTP POST
id samp	2	java.io.File: void <init< td=""><td>FILE</td><td>springframework.http.HttpEntity,java.lang.Class,java.lang.Object[])</td><td>NETWORK</td><td>request. There are checks along the flow for checking if the app Viber is installed.</td></init<>	FILE	springframework.http.HttpEntity,java.lang.Class,java.lang.Object[])	NETWORK	request. There are checks along the flow for checking if the app Viber is installed.
vibleaker andro		,				
id samp	3	iava.io.File: void <init< td=""><td>FILE</td><td>java.io.File: java.lang.String getName()</td><td>NETWORK</td><td>This malicious flow sends image to a remote server.</td></init<>	FILE	java.io.File: java.lang.String getName()	NETWORK	This malicious flow sends image to a remote server.
vibleaker andro		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		Jan 1		
id samp	4	java.io.File: void <init< td=""><td>FILE</td><td>java.io.File: java.lang.String getName()</td><td>NETWORK</td><td>This malicious flow sends images to a remote server.</td></init<>	FILE	java.io.File: java.lang.String getName()	NETWORK	This malicious flow sends images to a remote server.
			<u> </u>	org.mozilla.javascript.Function: java.lang.Object		This malicious partial ICC flow starts by reading the Intent with which the SMSHandler BroadcasReceiver
xbot android s		android.telephony.SmsMessage: android.telephony.SmsMessage		call(org.mozilla.javascript.Context,org.mozilla.javascript.Scriptable,org		started. The intent contains SMS messagees, which will be sent to a remote server with some script call
amp	1	createFromPdu(byte[])	SMS MMS	.mozilla.javascript.Scriptable,java.lang.Object[])	CRITICAL FUNCTION	defined by the malware writer.
	-				2	
						This malicious flow reads the device id and put it into an Array of String, the array is used as the return
xbot_android_s				android.webkit.WebView: void		value of the method getTelephonyInfo() of the class xAPI. An object of xAPI is injected into the WebView
	2	android.telephony.TelephonyManager: java.lang.String getDeviceId()	UNIQUE IDENTIFIER	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	by addJavascriptInterface. The method getTelephonyInfo() can be invoked via JavaScript code.
	Ē		J Zoe ibertiii ien		2	- 7 The method get deprion, may all be moded to Jurison pt code.
		android.content.ContentResolver: android.database.Cursor				This malicous reads contacts and store them into an ArrayList, this ArrayList is used as the return value
xbot android s		query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],jav		android.webkit.WebView: void		of the method getContacts() of the class xAPI.An object of xAPI is injected into the WebView by
	3	a.lang.String)	DATABASE	addJavascriptInterface(java.lang.Object,java.lang.String)	CRITICAL FUNCTION	addJavascriptInterface. The method getContacts() can be invoked via JavaScript code.
шһ	3	a.iang.ounig)	DATABASE	aucuavascriptinterracetjava.iang.Object,java.iang.string)	CRITICAL FUNCTION	adusavascriptimenace. The method getcontacts() can be invoked via savascript code.