# A RUNTIME TECHNIQUE FOR Identifying malicious applications

# HUNTING ANDROID MALWARE

### WHOAMI

- Chris Le Roy
- Security "Researcher/Engineer"
- Øbrompwnie



- Problem
- Question
- Idea
- PoC
- Results
- Conclusion

#### THE PROBLEM

### MALWARE HAS AND IS A CONSTANT THREAT IN THE ANDROID ECOSYSTEM

#### ars technicauk a leat teo solice pilly cars caning four

Malicious apps with >1 million downloads slip past Google defenses twice

Malware scanners fail to detect 50 apps that charged for fake services.

#### Found: New Android malware with never-before-seen spying capabilities

Skygofree is among the most powerful spy platforms ever created for Android. 2018

# Currency-mining Android malware is so aggressive it can physically harm phones

This is your phone on mining software. Any questions?

2017

DAN COCDIN - 12/19/2017, 8:40 PM

### PREVIOUS RESEARCH LED ME TO THIS RESEARCH

#### 

### HOW DO I PROTECT MYSELF FROM THESE KIND OF ATTACKS?

- We have to look at the APK
  - Statically
  - In a sandbox of sorts

# WE LOOK AT MALWARE IN A FEW WAYS

- Hashes
- Code Signatures
- Permissions Reputation
- Behaviour

# HOW ARE WE PROTECTED BY MALWARE?

- Google Play Protect
- Google Playstore
- Third-Party Software
  - Anti-Virus
  - OS Support
  - MDM's, MAM's

# WHAT'S THE SHORTCOMINGS OF EXISTING TECHNIQUES

- Static Analysis is hard
- Can't run Cuckoo on my phone
- Scalability
- What if the app is not on an official Store?
- Bypassing AV is too easy
- Forensics is cool but how do you do it at realtime?
- Static analysis can only reveal a subset of the app's functionality

### NO RELIABLE WAY TO DETECT MALWARE ON DEVICES

# BUT THERE IS HEAPS OF DATA TO BE LOOKED AT!

- Android apps make use of objects
- Import statements are useful BUT
- You can import but not instantiate
- If it's instantiated, something is using the object
- Instantiated objects have data (some)

## ALL THIS DATA HAS TO BE SOMEWHERE TO BE LOOKED AT

- /PROC/{PID}/MAPS?
  - Analyse the heap regions?
- Analyse Heap Dumps? (HPROFs)
- Memory Forensics?
  - LiME ~ Linux Memory Extractor
  - Volatility
- (gdb) x /20xg 0x7fbd6208?
  - myObject.hashCode()
  - Not too bad with the DVM (dlmalloc)

# INSTRUMENTATION

- Objects exist on the heap so they are accessible
- Trace calls and monitor/engage with behaviour
- Its relatively easy
- Great way to gain insight into applications
- Object carving functionality is AWESOME



# WOULDN'T IT BE COOL IF AT RUNTIME I COULD SEE

- Which objects an app is using
- Which objects are instantiated
- What are the values for these objects

### THIS WOULD GIVE ME AN IDEA AS TO WHAT AN APP IS DOING AND HOW

### FOR EXAMPLE, ANALYSING AN APP WITH A METERPRETER BACKDOOR:

- Experience tells me to look for:
  - DexClassLoader
    - And what this injected code does
  - TCP Connection
- Which tells me that this app is
  - Injecting code at runtime
    - For example, encryption/decryption routines
  - Communicating remotely

### **DEMO: BASIC MALWARE INFECTION**

		Ubuntu 84-bit	0	0.00		
Applications 👻 Places 👻 💽 Terminal 👻	Tue 12:24	N 1 / 0	ి 🕛 - 🚺	Nexus 5	• • 0	••• C O
root@	kaliBrompwnie: ~	0				T 1 12:24
File Edit View Search Terminal Tabs Help			E	Google		Ŷ
root@kaliBrompwnie: ~ ×	root@kaliBrompv	vnie: « 🗙	8 .			
<pre>sf exploit(multi/handler) &gt;</pre>	I			Twitter Twitter Togle Google Creat	•	Play Store
				4	0	
				<	0	=
		(8 12 12 19 19 19 19 19 19 19 19 19 19 19 19 19				

#### IN ACTION

### **DEMO: BASIC RUNTIME MALWARE ANALYSIS USING FRIDA**

•••	Demo"]_MalwareObjects.js — Kwetza	000	
ð	15 Demo1_MelwareObjects.js x		0 B 4 4) C ()
م	<pre>ver objectsToLookFor = ["java.net.Socket", "dalvik.system.DexClassLoader", "java.net.URLConnection", "java.net.URL", "java.securi for (var i in objectsToLookFor) {     Java.perform{function () {         [s         ]         [s         ]         ]         ]</pre>		🎔 🖹 🛧 🕅 12:32 1 Sign up
¥	<pre>3 Java.chooselobjectsToLookForli), { 4 "onMatch": function (instance) { 7 if (objectsToLookFor[i] "java.net.URL" 66 instance.getProtocol() != "file") { 7 } 7 } 7 } 7 } 7 } 7 } 7 } 7 } 7 } 7 }</pre>	Log in to Twitter.	
¥)	<pre>6 console.log("\n[+] Process has Instantiated instance of: " + objectsToLookFor[1]); 9 console.log("[*] Process is communicating via " + instance.getProtocol()); 10 console.log("[+] Communication Details: " + instance.toString());</pre>	Phone, email or usernam	2
¢.	11     J       12     if (objectsToLookFor[i] == "dslvik.system.DexClessLoader") {       13     console.log("\n[+] Process has Instantiated instance of: " + objectsToLookFor[i]);       14     console.log("[*] Process is making of DexClassLoader");	Brompwnie	
	<pre>13 conspleting("[+] Loader Details: " + instance.toString()]; 16 } 17 if (objectsToLookFor[1] -= "java.net.Socket") {</pre>	apassword	١
	18     console.log("\n +] Process has Instantiated instance of: " + objectsToLookFor[i]);       19     console.log("[*] Process is making of a Socket Connection");       20     console.log("[+] Socket Details: " + instance.toString(]);	Forgot p	assword?
	21     }       22     if (objectsToLookParli) == "java.met.URLConnection") {       23     console.log("\n[+] Process has Instantiated instance of: " + objectsToLookFor[i]);       24     console_log("\n[+] Represe is reking of a URL Connection");	1 2 3 4 5	Log in
	PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL f: bash + f ~ [] : cleroy-ltm:Scripts cleroy\$ frida -0 -l Demo1_MalwareObjects.js con.twitter.android	qwert	yuiop
		asdf	, hj kl
		1 z x c y	/bnm 🖾
		7123 , 💮	
		< (	) ≡
3		Ser Ser	

۲

#### STATIC VS RUNTIME ANALYSIS

## STATIC ANALYSIS WONT SHOW YOU EVERYTHING

- Runtime Injection
  - Class Loaders
    - Very powerful to inject your functionality at runtime, requires analyst to acquire the jar/apk/dex
  - What if you don't have the injected JAR/APK?
    - /data/data/com.app.sandbox
- Java.Lang
  - Runtime.exec("/bin/sh")
  - No Import Statements
  - Instantiated but kinda immutable

#### STATIC VS RUNTIME ANALYSIS

### **DEMO: WHAT STATIC ANALYSIS CAN'T SHOW YOU**

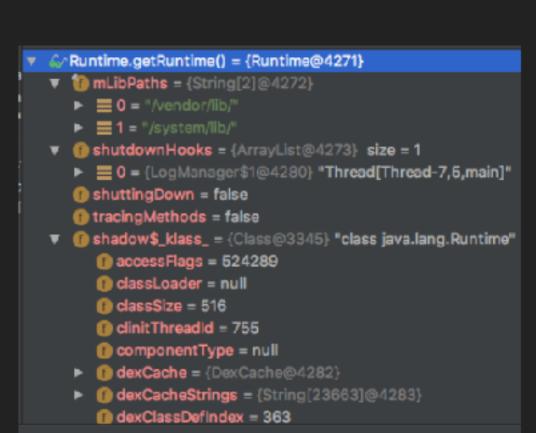
000	MainActivityJava - VictimAca - [-/AncheldStudioProjecta/VictimApe]	4	💿 💿 👘 LaadMajara - TheFundmakep - (H-VerdindeStackFrojecta)Tseiturstnekap)
By VielaA	er bij opp Mitten Mitmain mijnen Dit einter Dit brangerine Ditaren Dit einterapp @ Maintafelity		📱 ligtbabarlinstep lignep litter. Intenis imjen Attention Attenueren Atten Attenueren Glandin — 🔨 Statutetter 🕨 (+ + & ) 🕫 🗓 🗏 其 📬 🔍 🖿
1 il	atz nam ani 🗉 🕐 handstarty juni 🐑 🖉 Articity v 🕐 app 💈 🖓 bood tamatas 💈 🛞 attriys parts 🖓 🖓 grata parametes 🦄	📋 hange and redak per 🗧 📸 grant as an appendix personal see :	C 1 Contraction and C Contraction Allower C
£	Neinhertoxity (werdseffi) prochage win tim to respective com via Lingupg		o n
<b>R</b> (			
8 4	Lines vill endruddi var Bund Rez Lines vill andreddi support vill app. AppGemperMettizityy		a signer jara. is.Betforedhader)
	incert android.view.Wiewg		1 S BROTT 1478.16.1982/6ET.001 5 BROTT 1478.16.1892/5ET.001 74.17 BROTT 1478.16.1892/5ET.001 74.17 BROTT 148.16.1892/5ET.001
			er i lagort java.et.ittoR.cometion;
	<pre>import java.is.File; import java.is.Indeceptime; import java.iss.File(t.InvestignTargetEnertier;</pre>		I Apprt [Halet.BL]
1	impert java.lang.reflect.invocationTangetException; import java.lang.reflect.Method;		
8 5	inpert Galviko system Bodčiassiooder;		12 Setter Class Longe (
10 6	public class Meinentially seconds ApplumpetAntially (		14 public static void subdesed() ( In the static production of the static st
	WW(T)de		<pre>10 Eystem.ovt.prdetTe("EUCTION TO STREEGY * + Purvime_getFuntime().tedtring()) 27 D E</pre>
	protected void enfrecte(bundle savedinstance(tate) [ super.on[reste(savedinstance(tate)]		<pre>public static whit purScell/onward) {</pre>
	astContentYiewOk (ayout scill(ity_nain);		TO 1 System.out.println("FUHTHE TO STRING: " + Puntime.getPuntime(1.toString());
			22 Process Process + Aution_protection (communit "Yystem/bir/or");
	public void isoBoxClasses() threws holedhicldDeception, ClassRetFoundDeception, RiegelAccessEcoption, N FileD FileS = new FileD convents "Asta / Access[acces]/act/10/2/3010/07 % ListFileSD: Co.	wSuchheihedDoestian, EvocatianTergetDoestian (	2
	Lagrad New "Loadbacticoscs", may "Property to Leadbacticoscs"()		A Derrol_MalvareObjects.c  & Derrol_MalvareObjects.c  & Derrol_MalvareObjects.c
Ξ.	<pre>fam (File (Ale : / Ales) {     file(File) = new File(String, value0((this.getFilesDir())) </pre>		
	findi vest issuider claisioner = tev vestiassinaderifile.eetatsoirterttil, tmour.eetasoirterv	m(), mourysearchrist: mu(), Classicoscer.cetSystemClassicAudeol	2 var ediettsleutokkor = ("tavauret.socket", "salvikusvaten.bextlassionder", "isvauretuskionection", "isvauretuski, "tavauretuski", "savauseturit", "isvauretuskionection", "isvauretuski, "tavauretuski", "isvauretuski, "tavauretuski", "isvauretuski, "tavauretuski", "isvauretuski, "tavauretuski", "isvauretuski, "tavauretuski", "isvauretuski, "tavauretuski, "tavauretuski", "tavauretuski, "tavaure
	Final Next Wassioner (Wassioner = New Pexilassioner(File.etAbsolster(tr)), Impuls.etAbsolster(tr) Lap.et New Totambed access, Inner "Arthor fact to Long Classified" (Casting), Final Classified access), International Classified (Casting), International Classified), International Classified (Casting), International Classified), International Classified), International Classified), International Classified, International Classified), International Classified), International Classified), International Classified (Casting), International Classified), Intern	Instituting and the second sec	A Jave-per function 0 4
	Notical makalleard = classificand.getNotical instaliant");		2 29 5 Java-desseletjactaTelosetor[1], [ 2 5 5 5 Formington's function Lingtons) [
	nutcheard drawle; ( m(Q))		7 if [objectsTelesconfor[1] = "[www.set.UEL" 46 instance.getProtocol() = "file") {
	Section Function (Command on Clair(S)).000, getPetCodE, name: "runtion: Command") :		Consula_log("Ini+ Process has Instantiated instance.getProtocol()); consula_log("ini Process is communicating via " + instance.getProtocol());
	runShellCommand.involve( = mdil);		in consults (g) (i) the set of constraints (g) the state of constraint (g)
	<pre>feetual uridoneection = classTubled.getPethod( immerit/contextion"); uridoneection (invoke) = call);</pre>		2 III 1 ) 12 If (windstationation for (instance) for (instance) (i
	Lag.pt tog "leadbod"acced", mess "No are done running new functionality");		23 considering ("public Process has Environmentated Environment of a " + objects house Process (Ed.) (
			14 considering(*(+) Process to making of Perflamitation(***))     15 considering(*(+) Londer Details: * + Londer.teString());
	public void renStaff(Time view) throw EMException, InterruptedException {		17 if [etjactsTeleocdor[1] = "jwa.set.Sockst") {
	try ( LandDexClassoca(1)		PHOBLENS DUTFUT DEEUG CONSOLE TEMMENAL
	contra interpretation et 4     contra interpretation et 4		clerey lim Scripts clervy& fride = 0 = L Band_MatStaticCantShudve.js victim.brompunie.com.victimopp
Delug	L ave		
	Cando Delagar Sa E E E E E E E E E E E		
	Paras 🗄 🖠 Theoda - 4 🔤 Variables 👘 🚽	-/ 🞇 Watches -/	
		★ 6 #	
	Variables are not available		
	Frenez ero net evaliable	N+ wstehes	
1 x			
8 2			
1			
5			
	0 /2 Android Pauliter 🖬 D. Messeurer 😰 B. Ingres: 🚳 B. Pelver	C Evenius 🖬 Perrinel 🔳 Deale Decole	
Gracie	auto timanec la 955ma (Aminues apo)	15 dans 4228 LIT UTI-41 Canee econterb 🍾 🕘 🗔	🕘 🔲 🕻 👰 G 🛕 Ü 🖉 Linit, Coixa). Spacencik UIF-B Li Javadicipa. 40 (2011). 🤮 📑

# **HEAPS OF LOVE**

- Don't have to trawl code
- Identify specific anomalies

# THERE IS ALSO HEAPS OF FRUSTRATION

- java.lang.Runtime
- Kind of immutable?
- exec("/system/bin/ps")
  - Does not have much of a footprint



### WHATS THE PLAN?

exec(String command)

Executes the specified string command in a separate process.

exec(String[] cmdarray) Executes the specified command and arguments in a separate process.

exec(String[] cmdarray, String[] envp)
Executes the specified command and arguments in a separate process with the specified environment.

exec(String[] cmdarray, String[] envp, File dir)
Executes the specified command and arguments in a separate process with the specified environment and working directory.

exec(String command, String[] envp)
Executes the specified string command in a separate process with the specified environment.

exec(String command, String[] envp, File dir)

Executes the specified string command in a separate process with the specified environment and working directory.

```
Java.perform(function () {
    ver targetClass = Java.use("java.lang.Runtime");
    targetClass.exec.overload('java.lang.String').implementation = function (x) {
        console.log("[*] exec() got called!: "+x);
        return this.exec(x);
    };
    targetClass.exec.overload('ILjava.lang.String;').implementation = function (x) {
        console.log("[*] exec() got called!: "+x);
        return this.exec(x);
    };
    targetClass.exec.overload('java.lang.String', '[Ljava.lang.String;').implementation = function (x) {
        console.log("[*] exec() got called!: "+x);
        return this.exec(x);
    };
    targetClass.exec.overload('java.lang.String', '[Ljava.lang.String;').implementation = function (x,y) {
        console.log("[*] exec() got called X= "+x);
        console.log("[*] exec() got called X= "+x);
        console.log("[*] exec() got called Y= "+y);
        return this.exec(x,y);
    };
};
```

#### **OVERCOME SOME FRUSTRATION**

#### DEMO: OVERLOAD METHOD CALLS TO OBSERVE OTHERWISE HARD TO OBSERVE ANOMALIES

<pre>* DemodylumisgiensOctiongly xt Java.perferm(function () {</pre>
<pre>ver terpetClass = 3.9vs.use("jow.long.furties");  ver terpetClass.exec.overload('jow.long.furties");  ver terpetClass.exec.overload('jow.long.furties");  terpetClass.exec.overload('jow.long.furties');  terpetClass.exec.overload('jow.long.furties');  terpetClass.exec.overload('jows.long.furties');  terpetClass.exec.overload('lijows.long.furties');  terpetClass.exec.exerload('lijows.long.furties');  terpetClass.exec.exerload('lijows.long.furties');  terpetClass.exec.exerload('lijows.long.furties');  terpetClass.exec.exerload('lijows.long.furties');  terpetClass.exec.exerload('lijows.long.furties');  terpetClass.exec.exerload('lijows.long.furties'); } </pre>
<pre>tarsetClass.exec.overload('java.lang.String').implementation = function (x) {     centrole.tog('(=) same() got called : "-ax);     return limis.exec.(x);     };     targetClass.exec.overload('(Ljava.lang.String;').implementation = function (x) {         centrole.tog('(=) same() got called is "-ax);         return limis.exec(();     };     targetClass.exec.overload('java.lang.String;', '(Ljava.lang.String;').implementation = function (x,y) 4     centrole.tog('(=) same() got called X= '+a);     centrole.tog('(=) same() got called X= '+a);     return limis.exec((x,y);     };     targetClass.exec.overload('(Ljava.lang.String;', '(Ljava.lang.String;').implementation = function (x,y) 4     centrole.tog('(=) same() got called X= '+a);     return limis.exec((x,y);     };     targetClass.exec.overload('(Ljava.lang.String;', '(Ljava.lang.String;').implementation = function (x,y) {         targetCla</pre>
<pre>     Console.log:"[+] soac() got called: "-sc);     return this.exec(t);     };      targetClass.exec.overload('[Ljava.lang.String;').implementation = function (x,y) {</pre>
<pre></pre>
<pre>13 centrole.log("[+] exect) got called X= "+p); 13 centrole.log("[+] exect) got called Y= "+p); 14 return this.exec(*,r); 15 }; 15 }; 16 centrole.log("[1] pvs.lang.String;", "(Ljavs.lang.String;").implementation = function (x,y) { 17 centrole.log("[1] pvs.lang.String;", "(Ljavs.lang.String;").implementation = function (x,y) { 18 centrole.log("[1] pvs.lang.String;", "(Ljavs.lang.String;").implementation = function (x,y) { 19 centrole.log("[1] pvs.lang.String;", "(Ljavs.lang.String;").implementation = function (x,y) { 19 centrole.log("[1] pvs.lang.String;", "(Ljavs.lang.String;").implementation = function (x,y) { 19 centrole.log("[1] pvs.lang.String;", "(Ljavs.lang.String;").implementation = function (x,y) { 10 centrole.log("[1] pvs.lang.String;"].implementation = function (x,y) { 11 centrole.log("[1] pvs.lang.String;"].implementation = function (x,y) { 12 centrole.log("[1] pvs.lang.String;"].implementation = function (x,y) { 13 centrole.log("[1] pvs.lang.String;"].implementation = function (x,y) { 14 centrole.log("[1] pvs.lang.String;"].implementation = function (x,y) { 15 centrole.log("[1] pvs.lang.String;"].implementation = function (x,y) { 15 centrole.log("[1] pvs.lang.String;"].implementation = function (x,y) { 16 centrole.log("[1] pvs.lang.String;"].implementation = function (x,y) { 17 centrole.log("[1] pvs.lang.String;"].implementation = function (x,y) { 18 centrole.log("[1] pvs.lang.String;"].implementation = function (x,y) { 19 centrole.log([1] pvs.lang.String;"].implementation = function (x,y)</pre>
13 ); 13 13 4 C 1 P 3 targetClass.exec.ovarleed('[Ljava.lang.String:', '[Ljava.lang.String:'].implementation = function (s,s) {
a mile support support on point of the state (c, y);     22     console.log("(=) exec() got called Y= "+g);       a mile     23     return this.enec(c, y);
<pre>n, InvocationTerperException {     25     targetClass.exec.overload('[Ljava.lang.String:', 'java.lo.File').implementation = function (x,y,z) {         cancels.lag("[el exect] got called Xe "ea);         centrole.lag("[el exect] got called Xe "ea);         centrole.lag("[el exect] got called Ye "ea);         centrole.lag("[el exect] got called</pre>
null, Classicoder.petSystemClassicoder           29         cennole.log("[#] sxec() got called 2# "#0)]           30         return this.exec(x.v.z);           31         };           32         33           33         targetClass.exec.overload('java.lang.String', '[k]ava.lang.String)', 'java.do.*ile').implementation = function (x,y,z) (
31     censole.log!"(+) exec() got called X= "+=);       35     censole.log!"(+) exec() got called Y= "+=);       35     censole.log!"(+) exec() got called Z= "+=);       35     censole.log!"(+) exec() got called Z= "+=);
37 return (bils.exect(,,,,z); 30 ); 30
PROBLIMS OUTPUT DEBUG CONSOLE TERMINAL 21 basis
x ho Fiters T
starage r tas_wei sz-szise sz-szise sz-szise sz-szise sz-szise r r/ r r/ r r r r r r r r r r r r r

# WE HAVE THE ABILITY TO:

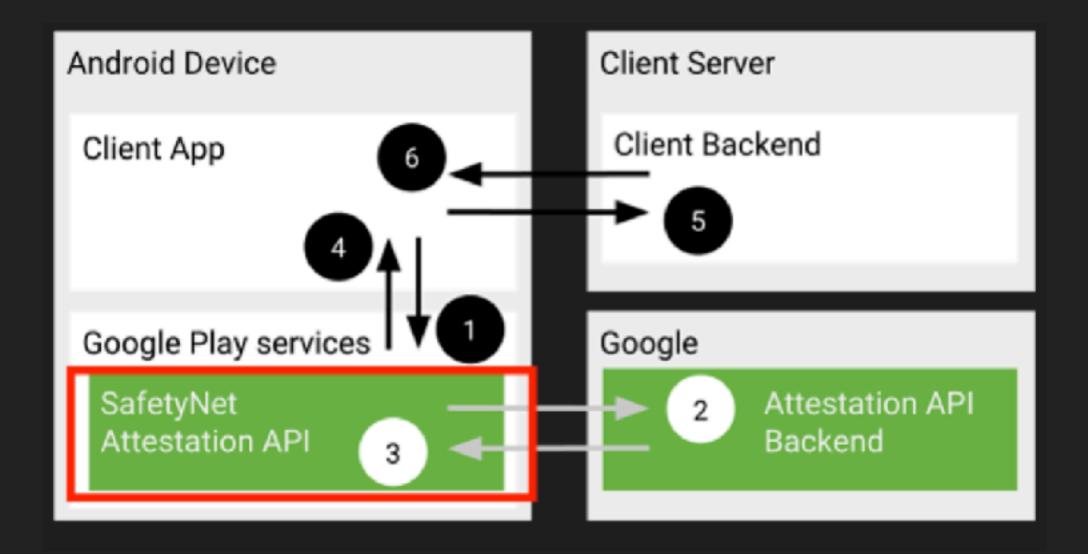
- Analyse objects on the heap
- Hook methods for certain objects
- Do all this at runtime on a device
- See more than static analysis
- Perform the above from a workstation

# A SOLUTION: SAFETY NET ATTESTATION API

The SafetyNet Attestation API helps you assess the security and compatibility of the Android environments in which your apps run. You can use this API to analyze devices that have installed your app.

#### HOW TO USE THIS?

## A SOLUTION: SAFETY NET ATTESTATION API



### UITKYK

- You can use this API to analyze applications that are installed on a Android device
- Custom Android Frida Library
- DBUS over TCP
- Frida Server Integration
- Can run all the previously demo'd tests
- > And more!

### HEY FRIDA, GIVE ME RUNNING PROCESSES

	\$ frida-ps -U
PID	Name
207	adbd
27599	android.process.acore
1566	android.process.media
16850	app_process
16894	app_process
17005	app_process
17145	app_process
17163	app_process
17190	app_process
17327	app_process
17384	app_process
17430	app_process
26767	app_process32
194	bridgemgrd
27556	com.android.defcontainer
27799	com.android.keychain
27622	com.android.musicfx
1799	com.android.nfc
1855	com.android.phone
1544	com.android.systemui
27643	com.android.vending
27882	com.google.android.apps.cloudprint

### HEY ANDROID, GIVE ME RUNNING PROCESSES

@Override protected String doInBackground(String... params) { UitkykUtils uitkykUtils= new UitkykUtils(); return uitkykUtils.fridaPS(fridaHost,fridaPort); 3

### HEY FRIDA TELL ME IF THIS APPLICATION LOOKS MALICIOUS

\$ frida -U -l Demo1\_MalwareObjects.js com.twitter.android



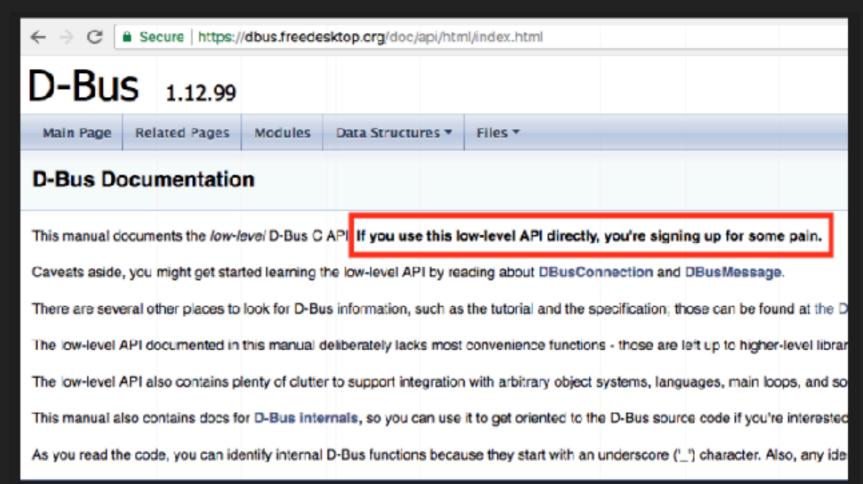
- [+] Socket Details: Socket[address=/192.168.0.16,port=4444,localPort=60695]
- [+] Process has Instantiated instance of: java.net.Socket
- [\*] Process is making of a Socket Connection
- [+] Socket Details: Socket[address=settings.crashlytics.com/50.19.106.12,port=443,localPort=42306]

### HEY ANDROID TELL ME IF THIS APPLICATION LOOKS MALICIOUS

protected String doInBackground(Object... objects) {
 UitkykUtils uitkykUtils = new UitkykUtils(fridaHost, fridaPort);
 return uitkykUtils.analyzeProcess(this.pid);

# WHY UITKYK API?

- No Android Frida Library
- Wanted to use Frida
- Wanted a Client Server Model
- Didn't want pain



# HOW DOES UITKYK UITKYK?

- TCP Socket to Daemon
- Push and Pull Bytes
- Sniffed Frida sessions
- Outlined TCP Flags
- Identified key bytes (trial and error)
- Stared at my monitor
- Wash,rinse,repeat

byte[]	AUTH_Me	ssage2	= -{					
	(byte)	0x41,	(byte)	0x55,	(byte)	0x54,	(byte)	0x48,
	(byte)	0x20,	(byte)	0x41,	(byte)	0x4e,	(byte)	0x4f,
	(byte)	0x4e,	(byte)	0x59,	(byte)	0x4d,	(byte)	0x4f,
	(byte)	0×55,	(byte)	0x53,	(byte)	0x20,	(byte)	0x34,
	(byte)	0x37,	(byte)	0x34,	(byte)	0x34,	(byte)	0x34,
	(byte)	0x32,	(byte)	0x37,	(byte)	0x35,	(byte)	0x37,
	(byte)	0x33,	(byte)	0x32,	(byte)	0x30,	(byte)	0x33,
	(byte)	0x30,	(byte)	0x32,	(byte)	0x65,	(byte)	0x33,
	(byte)	0x31,	(byte)	0x0d,	(byte)	0x0a};		

### **SOMETHING ABOUT A POC**

import socket

TCP\_IP = '10.42.0.15' TCP\_PORT = 1337 BUFFER\_SIZE = 100 # .AUTH MESSAGE = '\x2e\x41\x55\x54\x48\x8d\x8a' # MESSAGE = '\x41\x55\x54\x48\x8d\x8a'

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)
print "Sending-> .AUTH"
s.connect((TCP\_IP, TCP\_PORT))
s.send(MESSAGE)
deta = s.recv(BUFFER\_SIZE)
print "received data:", data
# s.close()

# AUTH ANONYMOUS 474442757320382e31
MESSAGE2="\x41\x55\x54\x48\x20\x41\x4e\x4f\x4e\x59\x4d\x4f\x55\x53\x20\x34\x37\x34\x34\x32\x37\x35\x37\x33\x32\x30\x33\x30\x32\;
BUFFER\_SIZE2 = 100
# s = socket.socket(socket.AF\_INET, socket.SUCK\_STREAM)
print "Sending-> AUTH ANONYMOUS 474442757320382e31"
# s.connect((TCP\_IP, TCP\_PORT))
s.send(MESSAGE2)
deta2 = s.recv(BUFFER\_SIZE2)
print "received data:", data2
# s.close()

# #
# #BEGIN
MESSAGE3="\x42\x45\x47\x49\x4e\x0d\x8a"
BUFFER\_SIZE3 = 100
# s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)
print "Sending-> BEGIN"
# s.connect((TCP\_IP, TCP\_PORT))
s.send(MESSAGE3)
# data3 = s.recv(BUFFER\_SIZE3)
# cata3 = s.recv(BUFFER\_SIZE3)

# print "received data:", data3
# # s.close()

# WHERE TO GET IT ALL

- Library
  - github.com/brompwnie/uitkyk
- Frida Scripts
  - github.com/brompwnie/uitkyk
- Videos
  - https://goo.gl/k6BNBq

### SHORTCOMINGS

- Increased Attack Surface
- Abuse, it is process running as root
- We are still struggling to get basic security right

# **CONCLUSION & QUESTIONS**

- It's a journey
- Uitkyk is a step in the right direction
- No Silver Bullet
- Defence In Depth
- Android OS is key to protecting itself
- Questions?