

Credits: <u>https://br.linkedin.com/in/atjunior</u> Frida Code: <u>https://codeshare.frida.re/@sowdust/</u>

When i started making pentesting on android applications, i had problems on intercepting requests with burp suite and other proxies.

So i've started to search more about SSL-Pinning.

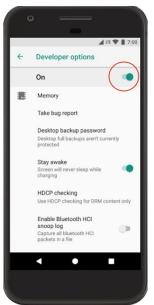
I made this tutorial using an Android Phone with ROOT and a Kali Linux Machine.

Let's start

First of all lets install ADB, Frida-Tools and <u>Burp Suite</u> on our machine. To install Frida-Tools we will need pip installed. To install pip: <u>Read</u>

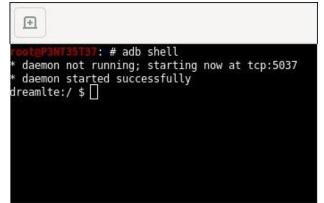
>> apt-get install adb
>> pip install frida-tools

After make this, we need to enable developer options on the phone:



With this option enabled, we conect our phone on the machine.

We go to our terminal, type *adb shell* Here is where we will make the conection with the phone, our phone will make an confirmation request, and we accept it.



Now we need to know the architecture of our phone because we will need to send the frida binary to change it into a "Android Server". To get the binary: <u>Github</u>

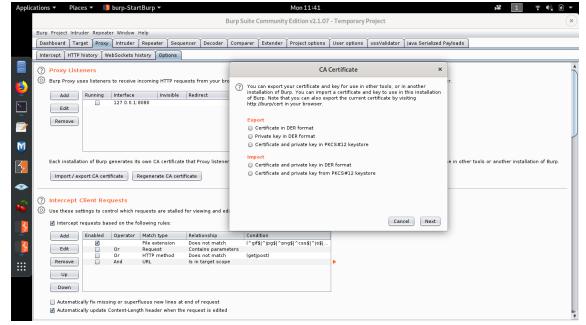
We need to send the frida binary using adb:

>> adb push frida-server-12.7.26-android-arm64 /data/local/tmp/

/data/local/temp/ \rightarrow is the directory of androd to copy

Applications 🔻	Places 🔻	›_ Terminal ▼	Mon 11:34
Ð			root@P3NT35T37: ~/Desktop/Frida
	r-12.7.26-ar	ndroid-arm <u>6</u> 4: 1 file pus	-server-12.7.26-android-arm64 /data/local/tmp/ hed. 17.1 MB/s (38480208 bytes in 2.152s)

Now we need to generate the Certificate of BurpSuite and send it to the same directory of frida binary file. On burp: We go on tab proxy > Tab options > Export > Certificate in DER format



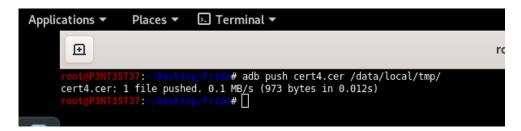
Now we need to change certificate extension from .der to .cer

I've created an certificate called cert4.cer, we can save it any place to send it to android phone.

Applic	ations 👻 Places 👻 🜗 burp-StartBurp 👻	Mon 11:42	
		Rurn Suite Community Edition v2.1.07 - Temporary Droject	×
	Burp Project Intruder Repeater Window Help	×	
	Dashboard Target Proxy Intruder Repeater		
	Intercept HTTP history WebSockets history Optic	Look jn: 📔 Frida 🔹 🔳 👰	
	Intercept HITP history WebSockets history Optic	Frida-gadget-lief-injector	-
	Proxy Listeners	adget-injector	-
-	Burp Proxy uses listeners to receive incoming HTT	p 🛅 lief-0.9.0.dev-57c30b8	
6		🕒 by.js	
_	Add Running Interface Invis	bypass-ssl.js cerl.js	
<u>}-</u>	Edit	cert.js	
_	Remove	Cert4.cer	
1		Frida-server-12.7.26-android-arm64	
_		frida_lief_injection.py	
M			
	Each installation of Burp generates its own CA cert		ther installation of Burp.
5			
	Import / export CA certificate Regenerate C		
~	(?) Intercept Client Requests		
. 10	Use these settings to control which requests are s		
-	Intercept requests based on the following rules		
3			
	Add Enabled Operator Match type	File Name: cert4.cer	
	Edit Or Request	Files of Type: All Files	
	Remove And URL		
		Save	
	Up		
	Down		
	Automatically fix missing or superfluous new ling	es at end of request	
	Automatically update Content-Length header wh		
			¥

Now we send it to the same directory of frida server on Android

>> adb push cert4.cer /data/local/tmp/



In terminal we'll run the binary we sent. So:

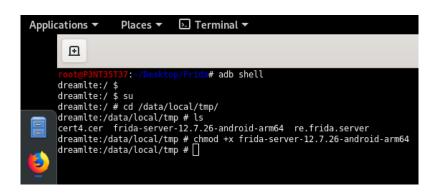
>> adb shell >> su

Su \rightarrow to root privileges

Ap	plications 🔻	Places 🔻	Ы Terminal 🔻	Mon 11:37
	Ð			root@P3NT35T37: ~/Desktop/Frida
	frida-serve root@P3NT35 dreamlte:/ dreamlte:/ root dreamlte:/	r-12.7.26-ar T37:~/Deskto \$ su # whoami		ida-server-12.7.26-android-arm64 /data/local/tmp/ pushed. 17.1 MB/s (38480208 bytes in 2.152s)

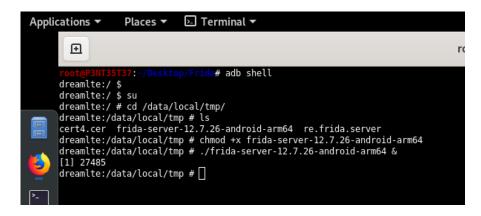
Now we need to change the privileges of binary execution

>> chmod +x frida-server-12.7.26-android-arm64



Lets run it:

>> ./frida-server-12.7.26-android-arm64 &



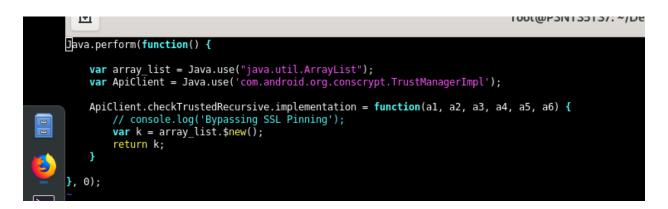
Now we need to see if everything is ok to start So we try to list android processes. Type:

>> frida-ps -U

Applic	cations	s 🔻 Places 🔻 🗈 Terminal 👻
	⊡	
	rooter	23NT35T37:~/Desktop/Frida# frida-ps -U
	PID	Name
	6608	adbd
	3406	android.hardware.audio@2.0-service
	3407	android.hardware.bluetooth@1.0-service
	3408	android.hardware.configstore@1.0-service
	3409	android.hardware.drm@1.0-service
	3410	android.hardware.drm@1.0-service.widevine
	3411	android.hardware.gatekeeper@1.0-service
	3467	android.hardware.gnss@1.0-service
>_	3413	android.hardware.graphics.allocator@2.0-service
	3128	android.hardware.keymaster@3.0-service

So we have frida and frida server running. Lets configure our proxy on android and burp. Create an .js to by-pass ssl-pinning

```
*/
Java.perform(function() {
    var array_list = Java.use("java.util.ArrayList");
    var ApiClient = Java.use('com.android.org.conscrypt.TrustManagerImpl');
    ApiClient.checkTrustedRecursive.implementation = function(a1, a2, a3, a4, a5, a6) {
    // console.log('Bypassing SSL Pinning');
    var k = array_list.$new();
    return k;
    }
    }, 0);
```



Recaping: we have burp, adb, pip, frida-tools, frida-server and burp certificate (.cer)

Lets configure it On android we go to connections > network management and we put our burp proxy: Mine: **IP: 172.108.2.249 port: 8080**

Applic	ation	ıs ▼ Pla	aces	•	burp-Sta	rtBurp 🔻					Mon 12:06					j u 1 후 •	
								В	urp Suite C	ommunity	Edition v2.1.0	- Temporary	Project				×
	Burg	Project Int	trude	r Repeat	er Window	Help											
	Da	shboard T	arget	Proxy	Intruder	Repeater	Seque	ncer Decoder	Comparer	Extender	Project options	User options	xssValidator	Java Serial	ized Payloa	ads	
	Int	ercept H∏	TP his	itory W	ebSockets h	istory 0	otions										
		Proxy Lis															1
	0	Burp Proxy	uses	listeners	s to receive	incoming				Edi	t proxy listener				×		
٢		Add	R	unning	Interface			Y	Certificat		e proxy ascene						
		Edit		đ	172.108.2	.249:	Binding	Request handling	Certificat	e					_		
<u>-</u>		Remove					? The	se settings control	how Burp b	inds the pro	xy listener.						
							Bine	d to port: 808	0								
2							Bin	d to address: 🔘 Lo	oopback only								
M						-	Dilli		ll interfaces								
		Each instal	llation	of Burp	generates it	s own C/		S	pecific addre	ss: 172.1	08.2.249		V		e in	other tools or another installation of E	urp.
5		Import /	expo	rt CA certi	ificate	Regenera											
						-											
	2	Intercept	t Cli	ent Req	uests												
2	0	Use these	settir	igs to con	trol which r	equests											
		🗹 Intercep	ot req	uests bas	sed on the f	ollowing											
3		Add	E	nabled	Operator												
-		Edit	5		Or	File ex Reque								ок Са	ncel		
		Remove	51		Or And	HTTP r		Is in target scope									
		_			0114	Uni		ia in turget scope									
		Up															
		Down															
								and of request									
		🗹 Automa	tically	rupdate (Content-Leng	gth header	when the	request is edited									

Android and Burp gotta be in the same network

On burp we can see if the requests have been intercepted

Appli	ations 🔻 🛛 Places 👻 📑 burp-StartBurp 👻 Mon 12:07
	Burp Suite Community Edition v2.1.07 - Temporary Project
	Burp Project Intruder Repeater Window Help
	Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options xssValidator Java Serialized Payloads
	Intercept HTTP history WebSockets history Options
	Forward Drop Intercept is on Action Con Raw Headers Hex
<u>-</u>	GET / HTTP/1.1 User-Agent: Dalvik/2.1.0 (Linux; U; Android 8.0.0; SM-G950F Build/R16NW) Host: 172.108.0.10 Connection: close Accept-Encoding: gzip, deflate
2	

Now the communication is working, we will try to by-pass Linkedin SSL because we are not intercepting application requests. So we need to list the processes again and select the app we will try to find bugs. Again:

>> frida-ps -U

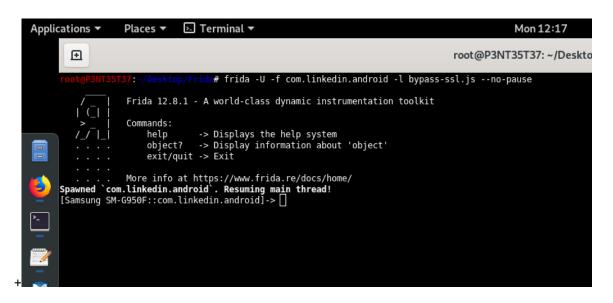
Now we locate Linkedin pid

	3440	apaservice
	3442	argosd
	3515	at distributor
		audioserver
	4992	
10		cameraserver
	3444	
	11089	com.android.bluetooth
	4574	com.android.nfc
4	4608	com.android.phone
7	2566	com.android.printspooler
_	4290	
		com.android.systemui:InfinityWallpaper
		com.android.vending
		com.android.vending:download_service
		com.android.vending:instant_app_installer
	2007	
	5406	com.google.android.gms
	4502	com.google.android.gms.persistent
	30051	com.google.android.googleguicksearchbox:interactor
	29953	com.google.android.googlequicksearchbox:search
	3150	com.google.process.gapps
	3520	com.linkedin.android
	5520	com, crinkeurn, androru

So we access the .js directory. Mine called: by-pass-ssl.js

>> frida -U -f com.linkedin.android -l bypass-ssl.js --no-pause

Note: Android will close and open automatically the process of Linkedin. At this time we bypassed it



Now on burp we just need to see the requests of HTTPS that have been intercepted by Linkedin app.

ations 👻 🛛 Places 👻 📑 burp-StartBurp 👻			Mon	12:19					1	
		Burp Sui	te Community Editi	on v2.1.07	- Tempo	rary Pro	oject			
Burp Project Intruder Repeater Window Help										
Dashboard Target Proxy Intruder Repeate	r Sequencer Decoder	Compar	er Extender Proje	ct options	User op	ions x	ssValidator	Java Serialized Payloads	1	
Site map Scope Issue definitions										
Filter: Hiding not found items; hiding CSS, image an	d general binary content;	hiding 4xx	responses; hiding em	pty folders						
Ai https://www.linkedin.com	Host	Method	URL	Params	Stat 🔺	Length	MIME type	Title	Comment T	ïme requ
▼ <mark> li</mark> ▼ ⊗ track	https://www.linkedin		/li/track		200	2354	text			2:18:51 6
	https://www.linkedin https://www.linkedin		/li/track?nc=157833	1	200	1909				2:19:01 6
f///iv]oU0 / ///y+6d!"õ!¥Ðf¥@!m§MSdb',	https://www.linkedin		/li/track?nc=157833 /li/track?nc=157833	~ ~	200 200	1845 1909				2:19:24 6 2:19:25 6
⊠ ∫∭İVkoâF …∫ý+Õ jU ~qö∬aÃ. ∥¤UUYcĭ	https://www.linkedin		/li/track?nc=157833		200	1909				2:19:34 6
	https://www.linkedin	POST	/sensorCollect/?acti	~	200	1908	JSON		1	2:19:25 6
▼ 🛅 sensorCollect ▼ 🐵 /	https://www.linkedin	GET	/sensorCollect/							
7										
_ ; ;;; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;										
		_				_	-			
	Request Response									
	Raw Params He	aders He								
	POST /li/track?nc=1		057 HTTP/1.1							
	Connection: close	_0m								
	Content-Length: 889									
	X-UDID: c9957309-a3									
	X-RestLi-Protocol-V		.0.0							
	Accept-Language: pt Csrf-Token: ajax:53		54347474							
	Content-Type: appli									
	X-LI-Track: {"osNam									
									","dpi":"xxxhdpi","devi	
								7a5e0c","timezoneOffse	t":-3,"storeId":"us_goo	gleplay",
	isAdTrackingLimited Content-Encoding: g		npName":"voyager-ar	101010-,-1	pversion		7.86"}			
	X-LI-Lang: pt BR									
	Cookie: JSESSIONID=	ajax:5375	L34895264347474; bc	ookie=v=:	2&79ab5fb	2-48a6-	4b10-898d-f	ff20f840fa61;		
								mubrEsSshWCb6; lang=v	=2⟨=pt_BR;	
	lidc=b=VGST00:g=188								75 0 0770 101)	
	Accept-Encoding: gz			U; Andro	ord 8.0.0	; pt_BR	; 5M-G950F;	: Build/R16NW; Cronet/	/5.0.3//0.101)	
	meccpic cheodany, 92.	ep, ucital								

Credits: <u>https://br.linkedin.com/in/atjunior</u> Frida Code: <u>https://codeshare.frida.re/@sowdust/</u>