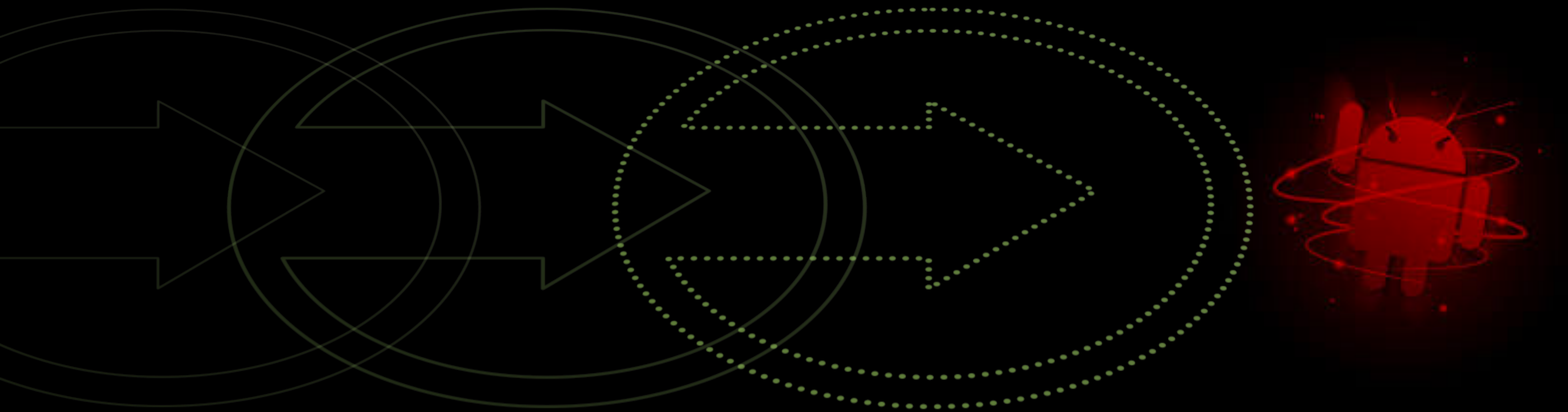
Android Persistent Threat



Application Testing & Assessment

Disclaimer/Format

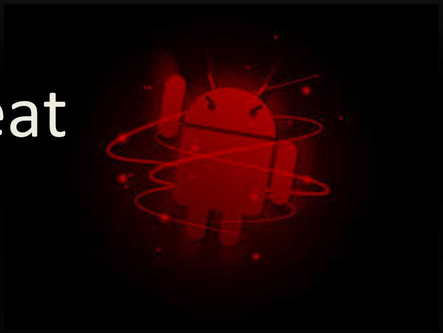
- This is a forum for learning and discussion – please “chime in”.
- The more I learn – the more I learn that I still have a LOT to learn, please don’t be afraid to ask questions!

- Please use this information responsibly - I don’t have enough bail money for all of us.

Attack Scenarios

- Man in the middle “coffee shop”
- Lost/Stolen device
- Trojaned application (omg, APT!)
- Malicious end user

- Android Persistent Threat



Attack Vectors

- Traffic analysis and injection
- Local device analysis and data modification
- Application reverse engineering and modification

- Each attack vector presents its own set of challenges

Things to consider: data

It's all about the data

- What's the intended purpose of the application?
- What's the data that the application processes?
  - How is that data classified? Is it sensitive? Should it be private?
  - Data doesn't need to be "TOP SECRET" in order to be classified - can be proprietary, or even personal.
  - Classification is the process of identifying your data and knowing what "type" it is.
- How *else* could this data be used?
- How is the data protected? In transit? At rest?
- What are the implications of modified data?
- How **ELSE** could this data be used?

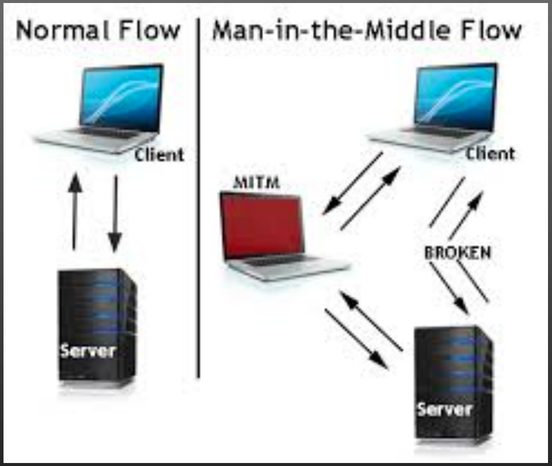
Other things to consider

It’s all about the lulz

- Using a physical mobile device vs emulator
  - Will vary depending on the threat you’re emulating
- Most mobile applications are just a “front-end” web application that talk over http/https
- Some mobile device management (MDM) solutions have encrypted containers for company content.
  - These are not always implemented correctly.

Man in the middle

Curious wireless packets



- You'll need to use a proxy
  - There are lots of ways to do this
- If you want to see encrypted traffic – you'll need to figure out how to defeat/get around that.

- My setup:
  - Burpsuite
  - Manually exported burp cert and installed on mobile
  - Force WIFI on mobile -> use burp for proxy
  - ???
  - lulz

Man in the middle

Pt. 2

- Things to look for:
  - Unencrypted credentials
  - Many apps rely on SSL (which we already bypassed).
  - Tokens/cookies
  - Base-64 encoded items
  - Database queries
    - Json, sql, etc.
  - Remote Command INJECTION*
  - Basically anything you'd look for on a web application.

*Thank you, @tothehilt for pointing out this important distinction between RCE and RCI

Local device analysis

Local filesystem inspection

- Will require ADB and some other tools – depending on what you're trying to accomplish
- Will need rooted device
- Can 'hide' root from applications if needed
    - Some "root-aware" apps just look for known binaries. ;]

Local device analysis

Local filesystem inspection pt. 2

- Tools:
  - android bridge, root, sqlite
- Some unencrypted things to look for:
  - Flat files used by the application
  - Databases
  - Cache files
  - Config files

Application Reverse Engineering

Application Analysis for Fun and Profit

- .apk files are a type of “zip” file that contain everything needed for an android app to run
- Need to decompile them in order to analyze/reverse engineer.
- Hard-coded values are common and can glean info.
    - API keys, internal IP space, comments, etc.
- Memory dumping/analysis is also possible – but not discussed in this presentation.

Tools:

Don't be a tool

- **Santoku Linux has most everything you should need –** https://santoku-linux.com
- Can piece tools together yourself, depending on needs and platform
- Android Studio
- Android bridge
- Apktool
- Lobotomy Framework
- Jd-gui
- Androguard
- MobSF*
- Lots more

Questions/Feedback?

Live demo time

I’m not your droid, buddeh!

Links

General
www.androidpentesting.com
https://santoku-linux.com
https://portswigger.net
https://www.rsaconference.com/writable/presentations/file_upload/stu-w02b-beginners-guide-to-reverse-engineering-android-apps.pdf
https://github.com/ajinabraham/Mobile-Security-Framework-MobSF

Other android attack vectors
https://www.rsaconference.com/writable/presentations/file_upload/mbs-f03-android-serialization-vulnerabilities-revisited.pdf
https://www.usenix.org/system/files/conference/woot15/woot15-paper-peles.pdf
https://www.usenix.org/sites/default/files/conference/protected-files/woot15_slides_peles.pdf