**Module 17: Hacking Mobile Platform**

**Attack Vector**

* OWASP Top10 mobile risks-2016
  + Improper platform usage
  + Insecure data storage
  + Insecure Communication
  + Insecure authentication
  + Insufficient crypto
  + Insecure authorization
  + Client code quality
  + Code tempering
  + Reverse engineering
  + Extraneous functionality
* Mobile attack vector
  + malware
  + data exfiltration
  + data tampering
  + data loss
* **SMS phishing (Smishing)**
* **Agent smith attack**
  + persuade the victim to install attacker’s app
  + replace legitimate app
  + produce a hugh volume of ads
* **SS7 vulnerabilitiy**
  + SS7 is a **communication protocol** that allows mobile users to exchange communication through another celular network
  + Operated depending on mutual trust between operators without any authentication
  + Exploit this vulnerability to perform a MITM
* **Simjacker:** SIM Card attack, a vulnerability associated with a SIM card’s S@T browser, a pre-installed software on SIM.

**Hacking Android OS**

* Include an OS, middleware, and key applications
* Android is a linux-based OS
* Android device administration API: provide **device administration features** at the system level. Allow developers to create **security-aware** apps that useful in enterprise setting.
* Android Rooting
  + Allow users to **attain privileged control**
  + Involve exploiting security vulnerabilities in the **device firmware** and copying the SU binary to a location in the current process’s PATH and granting it exetuable permission with the **chmod command**
  + Tools: **KingoRoot, One Click Root, TunesGo Root Android Tool**
* Blocking WIFI access using **NetCut**
* Identify attack surfaces using **drozer**
* Hacking with **zANTI** and **Network Spoofer**
* Launch DoS using **LOIC**
* Session hijacking using DroidSheep
* Hacking with **Orbot Proxy**: A proxy app that empowers other apps to privately use the Internet
* Exploiting android device through **ADB (Android Debug Bridge)** using **PhoneSploit**
  + ADB: Allow attackers to communicate with the target device
* Sniffer: FaceNiff
* Launch **MITD (Man in the disk)** attack: Lead to the installation of potential malicious app
* Launch spearphone attack: Allow apps to **record loudspeaker data** without privileges.
* Android trojans: **Gustuff, xHelper**
* Hacking tools: **cSploit, Fing-Network Tools**
* Security tools: **Kaspersky mobile av**
* Device tracking tools: **google find my device**
* Vulnerability scanners: **X-ray**
* Online Android analyzers: **Online APK analyzer**

**Hacking IOS**

* Jailbreaking IOS
  + The process of **installing a modified set of kernel patches** that allow users to run third-party apps not signed by the OS vendor
  + Provide root access
  + Remove **sandbox restrictions**
  + Types of jailbreaking
    - Userland exploit: Allow **user level access**
    - iBoot Exploit: Allow both user level access and iboot level access
    - Bootrom Exploit: Allow both user level access and iboot level access
  + Jailbreaking techniques
    - **Untethered jailbreaking:** In an untethered jailbreak, if the user turns the device off and back on, the device will start up completely and the kernel will be patched without the help of a computer; in other words, **the device will be jailbroken after each reboot.**
    - **Semi-tethered jailbreaking:** In a semi-tethered jailbreak, if the user turns the device off and back on, the device will start up completely. It will no longer have a patched kernel, but it will still be usable for normal functions. To use jailbroken addons, the user needs to start the device with the help of the jailbreaking tool.
    - **Tethered jailbreaking:** With a tethered jailbreak, if the device starts up on its own, it will no longer have a patched kernel, and it may get stuck in a partially started state; to start it completely and with a patched kernel, it essentially must be “re-jailbroken” with a computer (using the “boot tethered” feature of a jailbreaking tool) each time it is turned on.
    - **Semi-untethered Jailbreaking:** A semi-untethered jailbreak is similar to a **semi-tethered jailbreak**. In this type of jailbreak, when the device reboots, the kernel is not patched. However, the kernel can be patched without using a computer; it is patched using an app installed on the device.
  + Jailbreaking IOS 13.2 using **Cydia**
  + Jailbreaking IOS 13.2 using **Hexxa Plus**
* Tools: **Apricot**, a web-based mirror operating system for all the latest iphones
* Hacking using **Spyzie**
* Hacking network using **Network Analyzer Pro**
* **IOS trustjacking:** A vulnerability that can be exploited to read messages and emails and capture sensitive info from a remote location without the victim’s knowledge. Exploit the “ITunes WIFI Sync” feature, where the victim connects their phone to any trusted computer that is already infected by an attacker
* Malware: **Clicker Trojan malware, Trident**
* Hacking tools: **Elcomsoft Phone breaker**
* Security Tools: **Avira mobile security**
* Tracking tools: **Find my iphone**

**Mobile Device Management (MDM)**

* Solutions: **IBM MaaS360, Citrix Endpoint Management**
* **BYOD**: Bring your own device is a policy that…
* BYOD Policy implementation:
  + Define requirements
  + Select the device and build a technology portfolio
  + Develop policies
  + Security
  + Support

**Mobile Security Guidelines and Tools**

* OWASP Top10 Mobile Controls
  + Identify and protect sensitive data on the mobile device
  + Handle password credentials securely on the device
  + Ensure sensitive data are protected in transit
  + Implement user authentication, authorization, and session management correctly
  + Keep the backend APIs (services) and platform (server) secure
  + Secure data integration with third-party services and applications
  + Pay specific attention to the collection and storage of consent for the collection and use of the user’s data
  + Implement controls to prevent unauthorized access to paid-for resources
  + Ensure secure distribution /provisioning of mobile apps
  + Carefully check any runtime interpretation of code for errors
* Reverse Engineering Mobile app
* Source code analysis tools: **z3A Advanced App Analysis**
* Reverse Engineering Tools: **Apktool**
* App repackaging detector
  + repackaging is the process of **extracting details of an app** from legitimate app stores
  + **Promon Shield**
* Protection tools: **Lookout Personal, Zimperium’s zIPS, BullGuard Mobile Security**
* Anti-spyware: **Malwarebytes for Android**
* Pentesting toolkit: **ImmuniWeb MobileSuite**