Hacking Mobile Platforms

Module 16

Ethical Hacking and Countermeasures v8

Module 16: Hacking Mobile Platforms

Exam 312-50

Security News

Mobile Malware Cases Nearly Triple in First Half of 2012, Says NetQin

Source: http://www.computerworld.com

In June, 3.7 million phones worldwide became infected with malware, Beijing researcher finds.

Mobile malware is rising fast, infecting nearly 13 million phones in the world during the year first half of 2012, up 177% from the same period a year ago, according to Beijing-based security vendor NetQin.

In a report detailing the world's mobile security, the company detected a major spike in **malware cases** in June, with about 3.7 million phones becoming infected, a historic high. This came as the security vendor found 5,582 malware programs designed for Android during the month, another unprecedented number for the period.

During this year's first half, NetQin found that most of the detected malware, at 78%, targeted smartphones running Android, with much of the **remainder** designed for handsets running Nokia's Symbian OS. This is a reversal from the same period a year ago, when 60% of the detected mobile malware was designed for Symbian phones.

In total, NetQin detected 17,676 **mobile malware** programs during 2012's first half, up 42% from the previous six months in 2011.

About a quarter of the detected malware came from China, which led among the world's countries, while 17% came from Russia, and 16.5% from the U.S.

In China, malware is mainly spread through forums, ROM updates, and third-party app stores, according to NetQin. So-called "remote control" **Trojan malware** that sends spam ads infected almost 4.7 million phones in China.

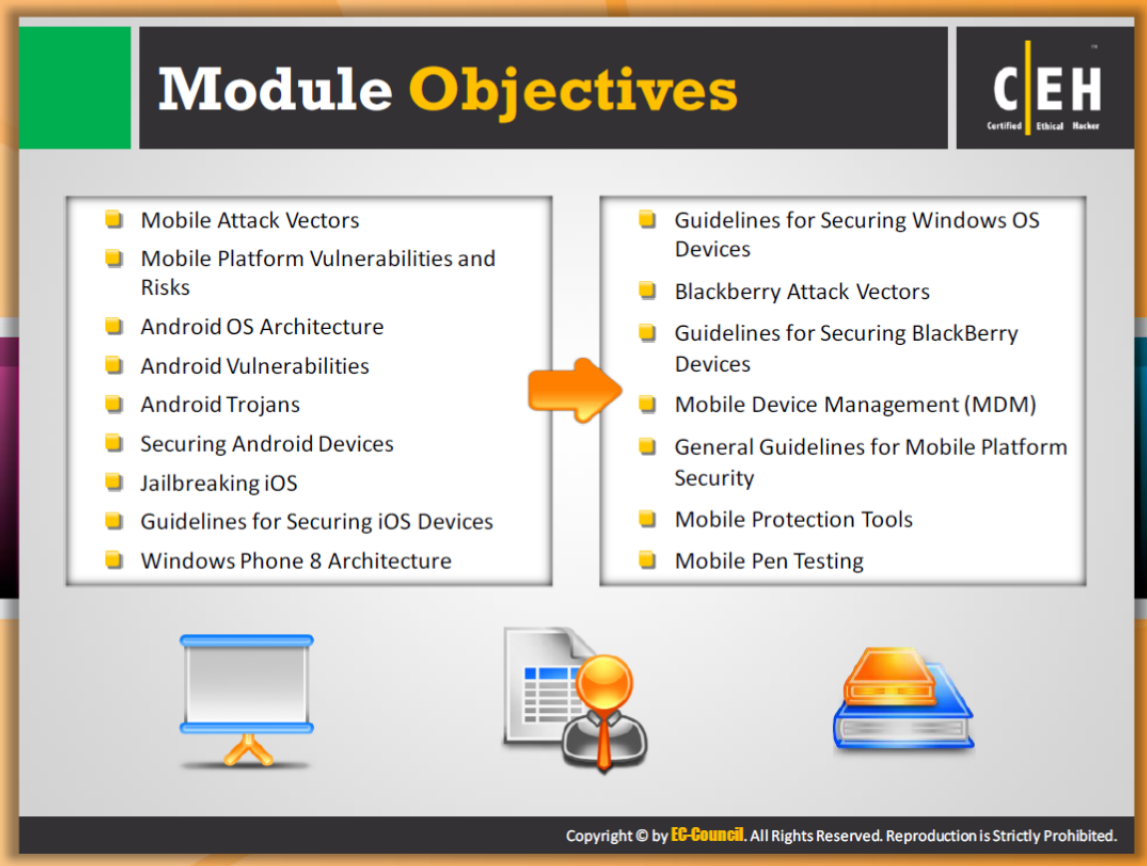
NetQin also detected almost 3.9 million phones in China being infected with money-stealing malware that sends out text messages to trigger fee-based mobile services. The high number of infections would likely translate into the malware's creators netting $616,533 each day.

The surge in mobile malware has occurred at the same time that China has become the world's largest smartphone market by **shipments**. Android smartphone sales lead with a 68% market share, according to research firm Canalys.

The country's Guangdong and Jiangsu provinces, along with Beijing, were ranked as the three highest areas in China for mobile malware.

By Michael Kan

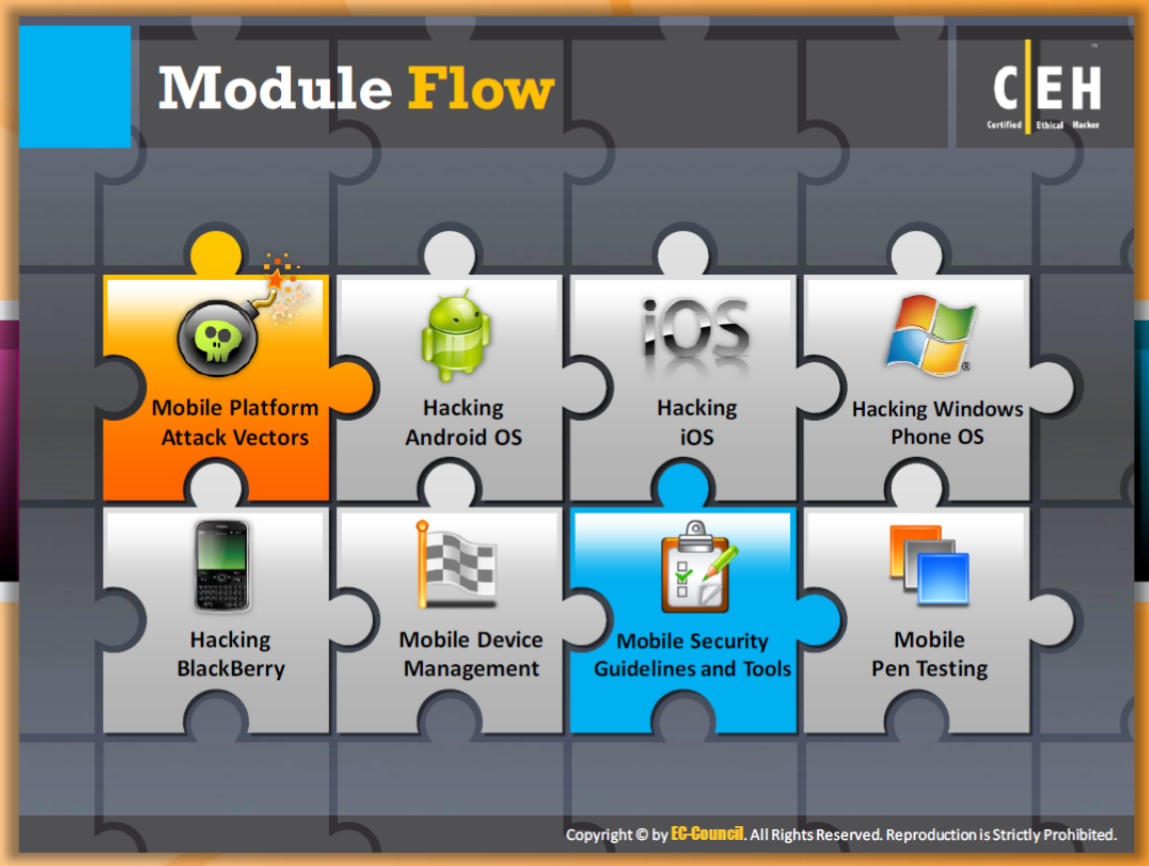
http://www.computerworld.com/s/article/9229802/Mobile_malware_cases_nearly_triple_in_first_half_of_2012_says_NetQin



Module Objectives

The main objective of this module is to educate you about the potential threats of mobile platforms and how to use the mobile **devices securely**. This module makes you familiarize with:

- Mobile Attack Vectors
- Mobile Platform Vulnerabilities and Risks
- Android OS Architecture
- Android Vulnerabilities
- Android Trojans
- Securing Android Devices
- Jailbreaking iOS
- Guidelines for Securing iOS Devices
- Windows Phone 8 Architecture
- Guidelines for Securing Windows OS Devices
- Blackberry Attack Vectors
- Guidelines for Securing BlackBerry Devices
- Mobile Device Management (MDM)
- General Guidelines for Mobile Platform Security
- Mobile Protection Tools
- Mobile Pen Testing

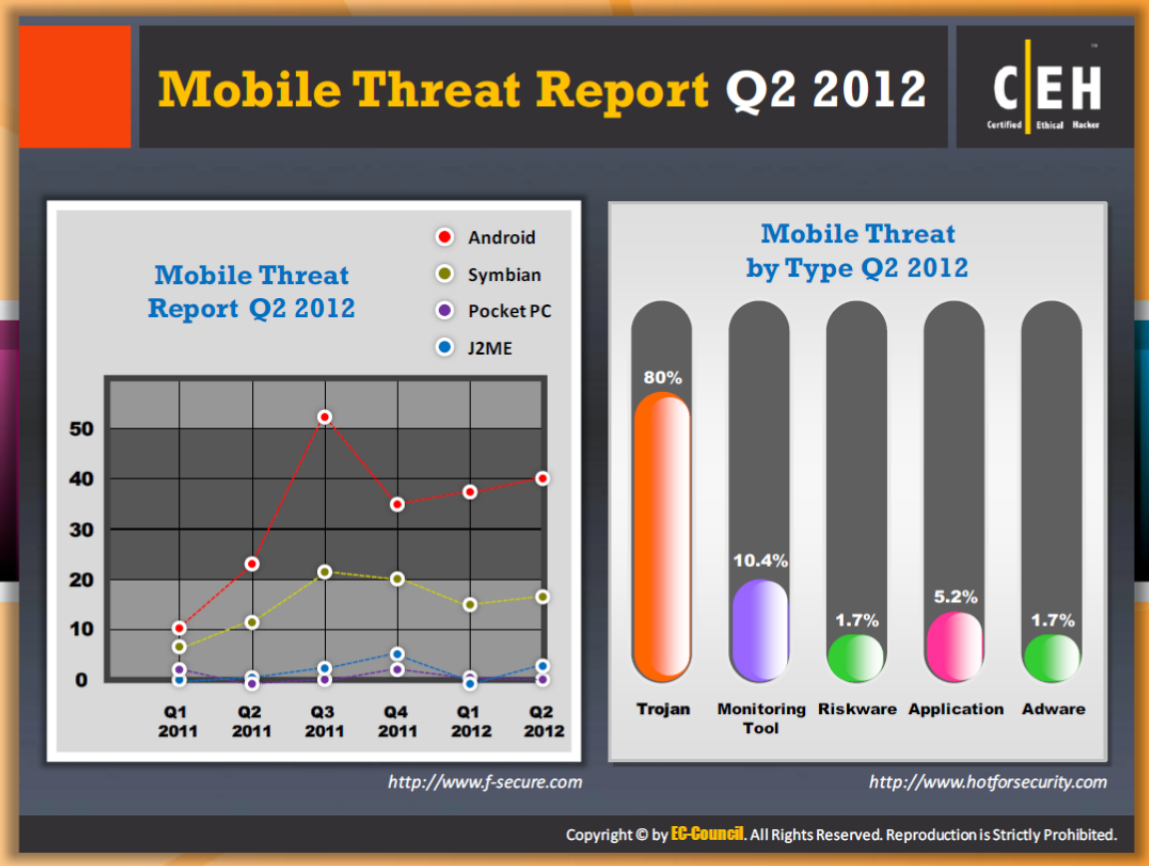


Module Flow

For better understanding, this module is divided into various sections and each section deals with a different topic that is related to hacking mobile platforms. The first section deals with mobile platform attack vectors.

Mobile Platform Attack Vectors	Hacking BlackBerry
Hacking Android iOS	Mobile Device Management
Hacking iOS	Mobile Security Guidelines and Tools
Hacking Windows Phone OS	Mobile Pen Testing

This section introduces you to the various mobile attack vectors and the associated vulnerabilities and risks. This section also highlights the security issues arising from app stores.



Mobile Threat Report Q2 2012

Source: http://www.f-secure.com

In the report, malware attacks on Android phones continue to dominate the other mobile platforms. The most attacks were found in the third quarter of 2011. And in 2012, Q2 came in at 40%.

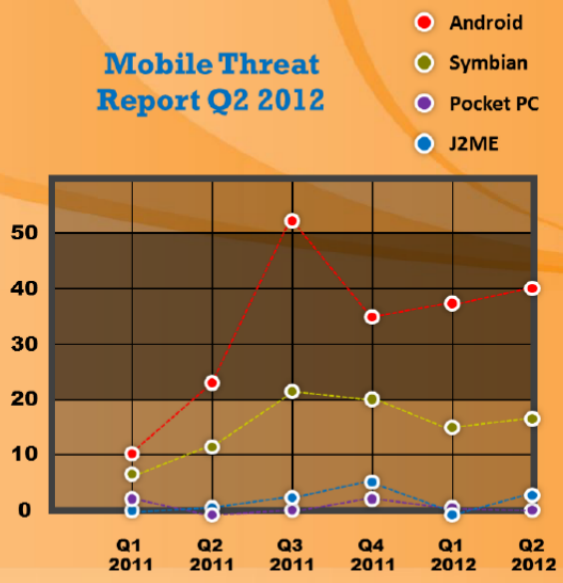


FIGURE 16.1: Mobile Threat Report Q2 2012

Note: The threat statistics used in the mobile threat report Q2 2012 are made up of families and variants instead of unique files.

Mobile Threat by Type Q2 2012

Source: http://www.hotforsecurity.com

Attacks on mobile phones were mostly due to the Trojans, which according to the Mobile Threat by Type Q2 2012. is about 80%. From the graph or report it is clear the major threat associated with mobile platforms is Trojan when compared to other threats such as monitoring tools, riskware, application vulnerabilities, and adware.

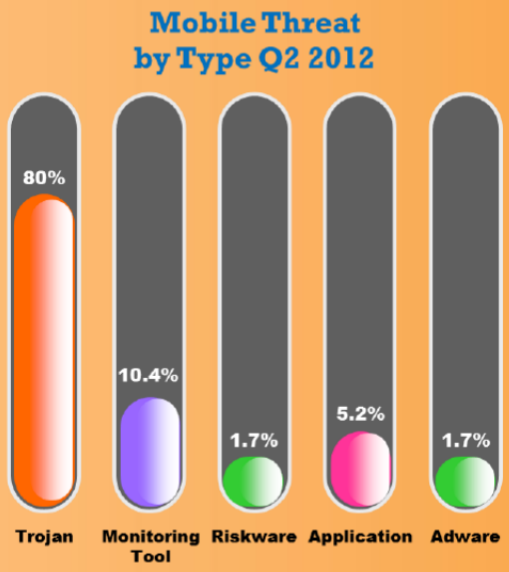
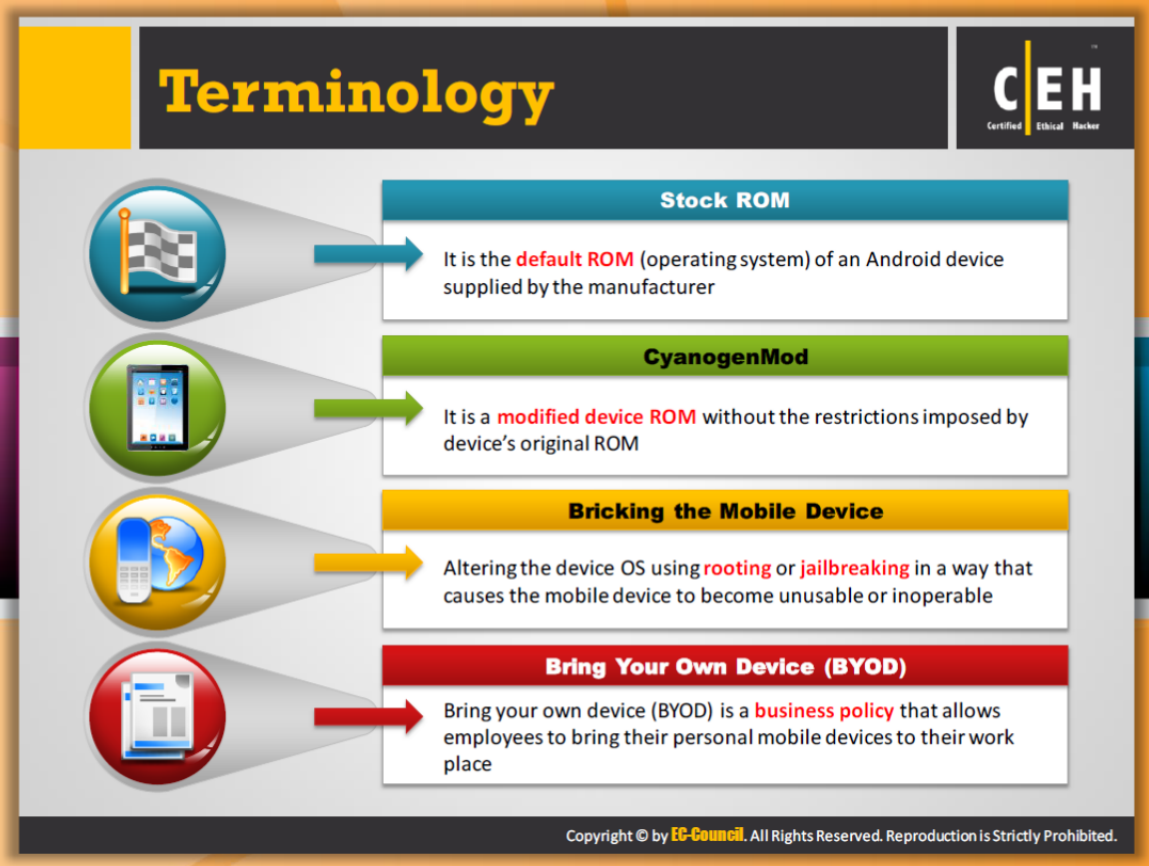


FIGURE 16.2: Mobile Threat by Type Q2 2012



Terminology

The following is the basic terminology associated with mobile platform hacking:

- **Stock ROM**: It is the default ROM (operating system) of an android device supplied by the manufacturer
- **CyanogenMod**: It is a modified device ROM without the restrictions imposed by device's original ROM
- **Bricking the Mobile Device**: Altering the device OSes using rooting or jailbreaking in a way that causes the mobile device to become unusable or inoperable
- **Bring Your Own Device (BYOD)**: Bring your own device (BYOD) is a business policy that allows employees to bring their personal mobile devices to their work place

Mobile Attack Vectors

Similar to traditional computer systems, most modern mobile devices are also prone to attacks. Mobile devices have many potential attack vectors using which the attacker tries to gain unauthorized access to the mobile devices and the data stored in or **transferred** by the device. These mobile attack vectors allow attackers to exploit the vulnerabilities present in **operating systems** or **applications** used by the mobile device. The attacker can also exploit the human factor. The various mobile attack vectors include:

Malware:

- Virus and rootkit
- Application modification
- OS modification

Data Exfiltration:

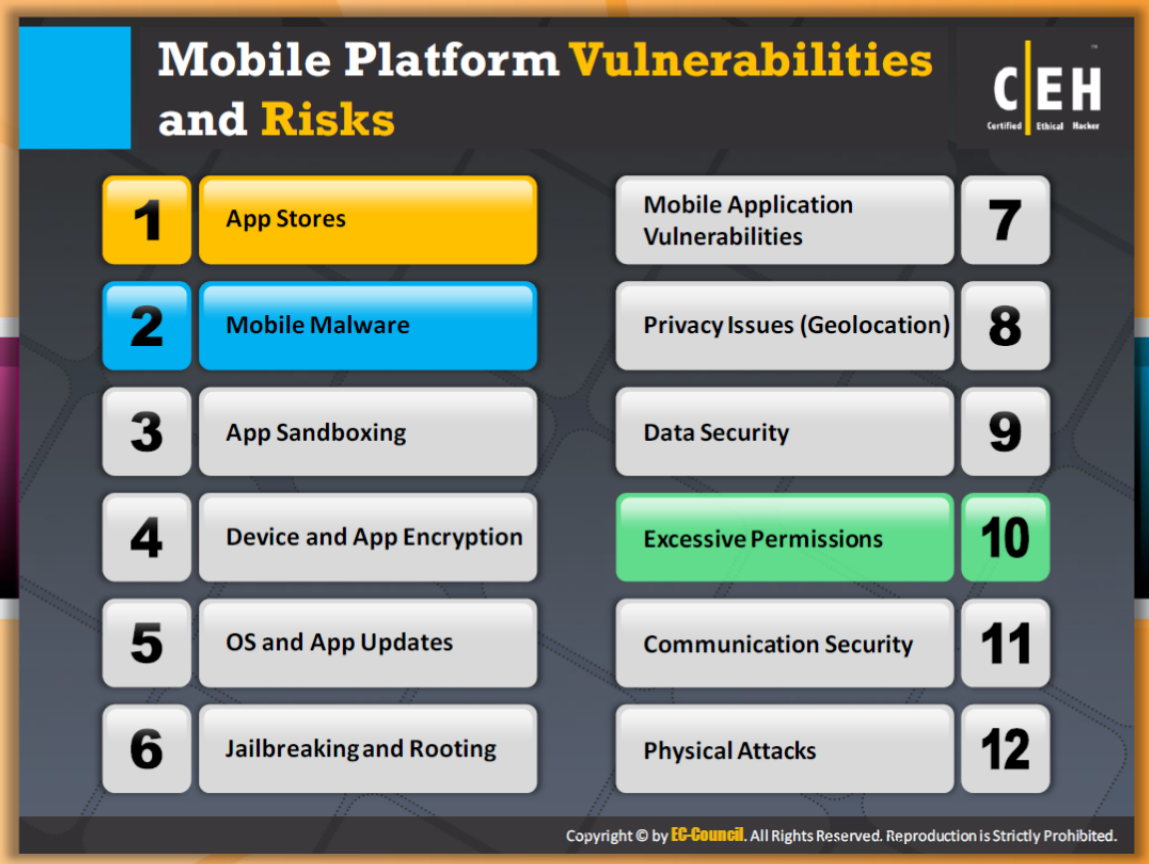
- Data leaves organization and email
- Print screen and screen scraping
- Copy to USB key and loss of backup

Data Tampering:

- Modification by another application
- **Undetected tamper** attempts
- Jail-broken device

Data Loss:

- Application vulnerabilities
- **Unapproved** physical access
- Loss of device



Mobile Platform Vulnerabilities and Risks

Mobile platform vulnerabilities and risks are the challenges faced by mobile users due to the functionality and increasing use of mobile devices at work and in other daily activities. The new functionalities amplify the attraction of the platforms used in mobile devices, which provide an easy path for **attackers** to launch attacks and exploitation. Attackers use different technologies such as Androids and other multiple instances to insert **malicious applications** with hidden functionality that stealthily gather a user's sensitive information. The companies that are into developing mobile applications are more concerned about security because **vulnerable applications** can cause damage to both parties. Thus, levels of security and data protection guarantees are mandatory. But the assistances and services provided by mobile devices for secure usage are sometimes neutralized by fraud and security threats.

The following are some of the risks and vulnerabilities associated with mobile platforms:

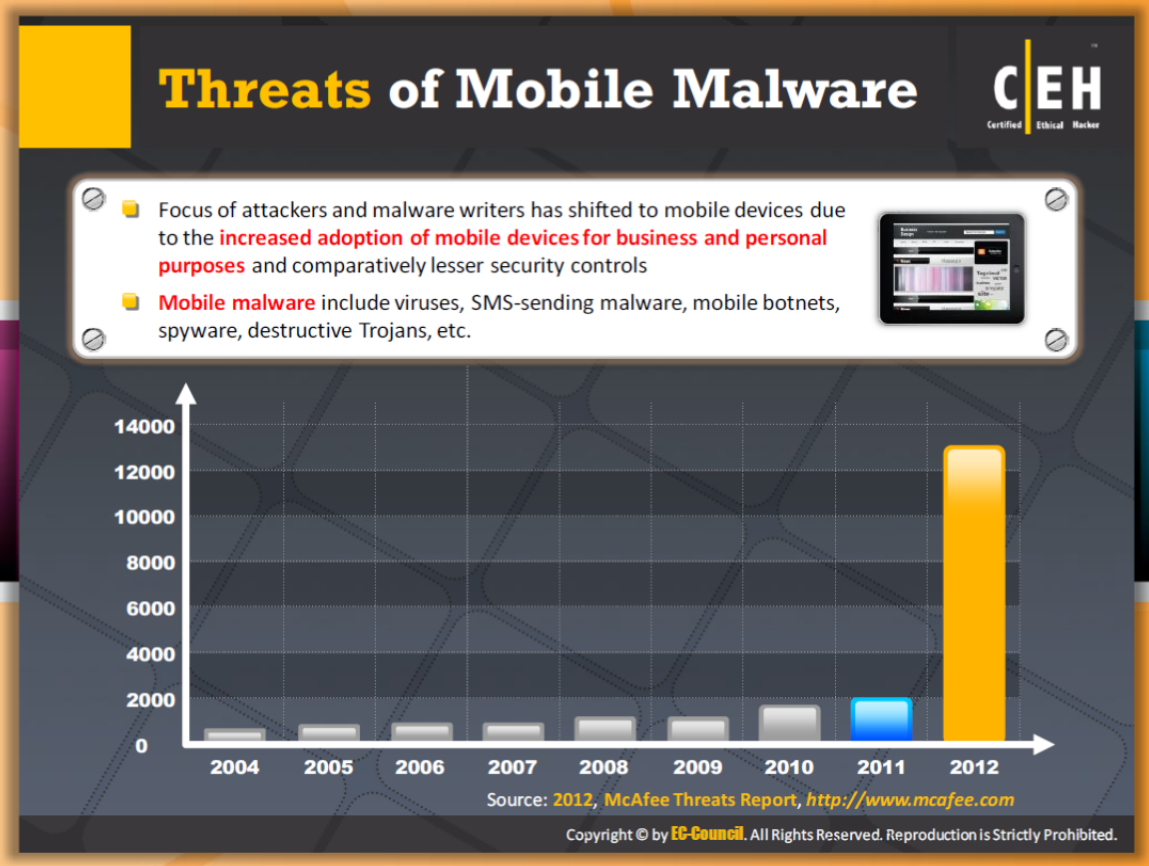
- App Stores
- Mobile Malware
- App Sandboxing
- Device and App Encryption
- OS and App Updates

- Jailbreaking and Rooting
- Mobile Application Vulnerabilities
- Privacy Issues (Geolocation)
- Data Security
- **Excessive Permissions**
- Communication Security
- Physical Attacks

Security Issues Arising from App Stores

An authenticated developer of a company creates **mobile applications** for mobile users. In order to allow the mobile users to conveniently browse and install these mobile apps, platform vendors have created **centralized** marketplaces, but security concerns have resulted. Usually mobile applications that are developed by developers are submitted to these marketplaces (official app stores and third-party app stores) without screening or vetting, making them available to thousands of mobile users. If you are downloading the application from an official app store, then you can trust the application as the hosting store has vetted it. However, if you are **downloading** the application from a third-party app store, then there is a possibility of downloading malware along with the application because third-party app stores do not vet the apps. The attacker downloads a **legitimate game** and repackages it with malware and uploads the mobile apps to a third-party application store from where the end users download this malicious gaming application, believing it to be genuine. As a result, the malware gathers and sends user credentials such as call **logs/photo/videos/sensitive** docs to the attacker without the user's knowledge. Using the information gathered, the attacker can exploit the device and launch many other attacks. Attackers can also socially engineer users to download and run apps outside the official app stores. Malicious apps can damage other applications and data, and send your sensitive data to attackers.

FIGURE 16.3: Security Issues Arising from App Stores



Threats of Mobile Malware

In recent years, many system users are moving away from using personnel computers toward smartphones and tablets. This increased adoption of mobile devices by users for business and personal purposes and comparatively **lesser security controls** has shifted the focus of attackers and malware writers for launching attacks on mobile devices. Attackers are attacking mobile devices because more sensitive information is stored on them. SMS spoofing, toll frauds, etc. are **attacks** performed by attackers on mobile devices. Mobile malware include viruses, SMS-sending malware, mobile botnets, spyware, destructive Trojans, etc. The malware is either application or functionality hidden within other application. For infecting mobile devices, the malware writer or attacker develops a malicious application and publishes this application to a major application store and waits until users install these malicious mobile applications on their mobile devices. Once the user installs the application hosted by the attacker, as a result, the attacker takes control over the user's mobile device. Due to mobile malware threats, there may be loss and theft, data communication interruption, exploitation and misconduct, and direct attacks.

According to the **threats report**, the security threats to mobile devices are increasing day by day. In 2004, malware threats against mobile devices were fewer when compared to recent years. The frequency of malware threats to mobile devices in the year **2012 drastically increased**.

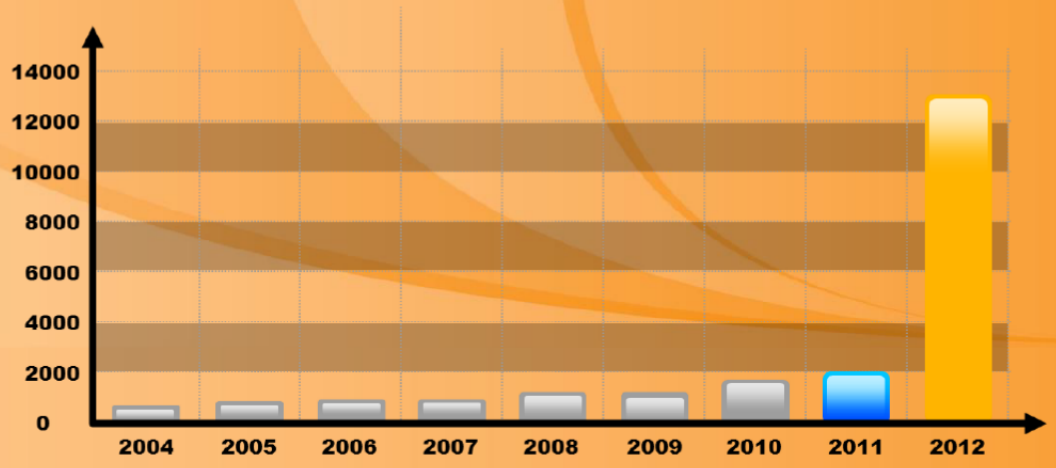
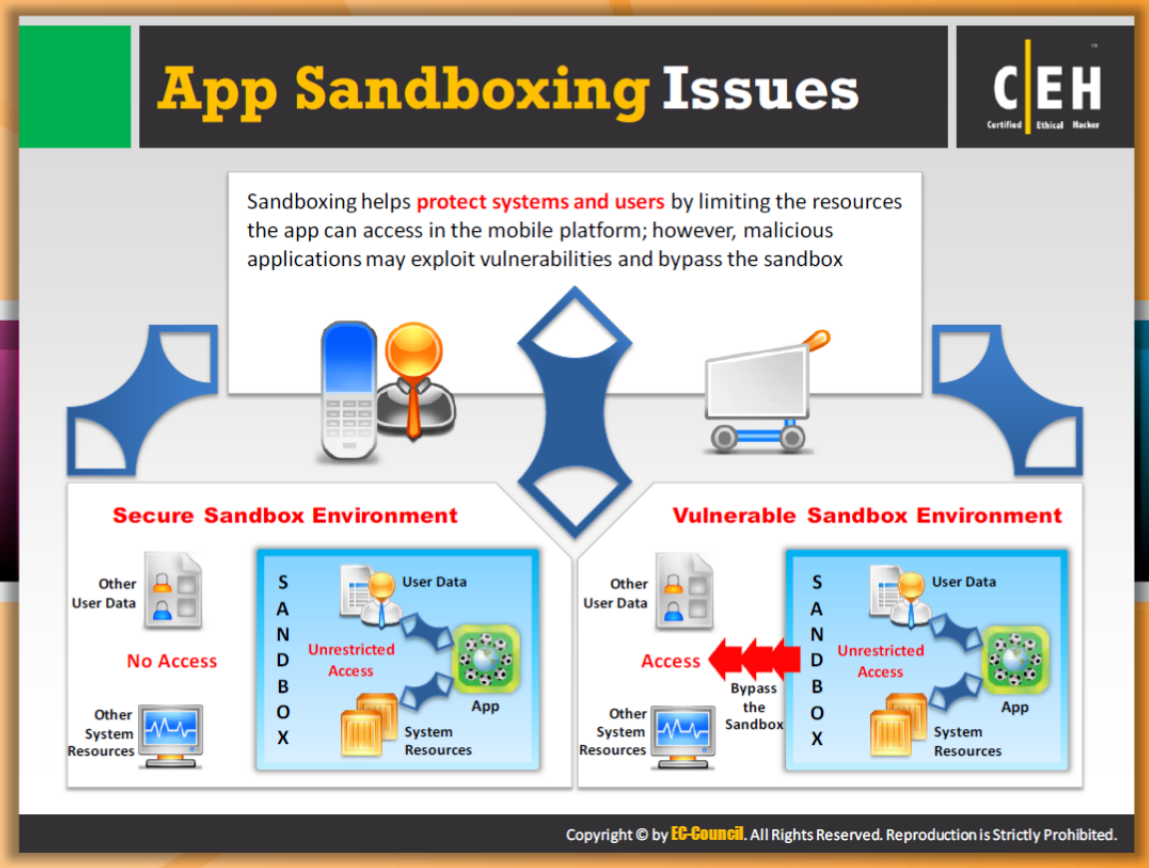


FIGURE 16.4: Threats of Mobile Malware



App Sandboxing Issues

Sandboxing separates the **running program** with the help of a security mechanism. It helps protect systems and users by limiting the resources the app can access in the mobile platform; however, malicious applications may exploit vulnerabilities and bypass the sandbox.

Sandboxing is clearly explained by comparing a computer and a smartphone. In normal computers, a program can access any of the system resources such as entire RAM i.e. not protected, hard drive information, and more can be read easily by anyone, unless and until it is locked. So if any individual downloads **malicious** software believing it as genuine, then that software can read the keystrokes that are typed in your system, scan the entire hard drive for useful file types, and then send that data back through the network. The same occurs in mobile devices; if an application is not given a working environment, it accesses all the user data and all the system resources. If the user downloads a malicious application, then that application can access all the **data** and **resources** and can gain complete control over the user's mobile device.

Secure sandbox environment

In a secure sandbox environment, each individual application is given its own working environments. As a result, the application is restricted to access the other user data and system resources. This provides **protection** to mobile devices against malware threats.

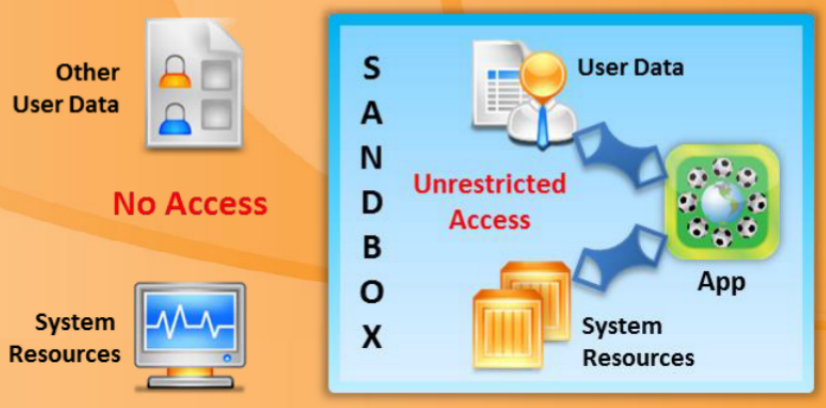


FIGURE 16.5: Secure sandbox environment

Vulnerable Sandbox Environment

In vulnerable sandbox environment, the malicious application exploits loopholes or weaknesses for **bypassing** the sandbox. As a result, the application can access other user data and system resources that are restricted.

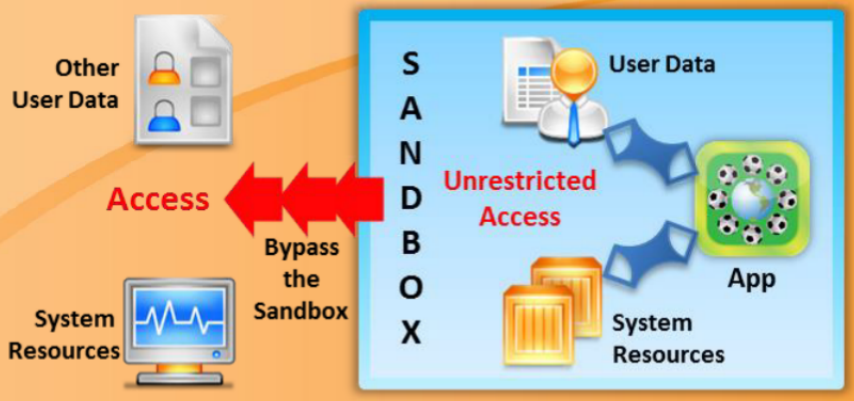
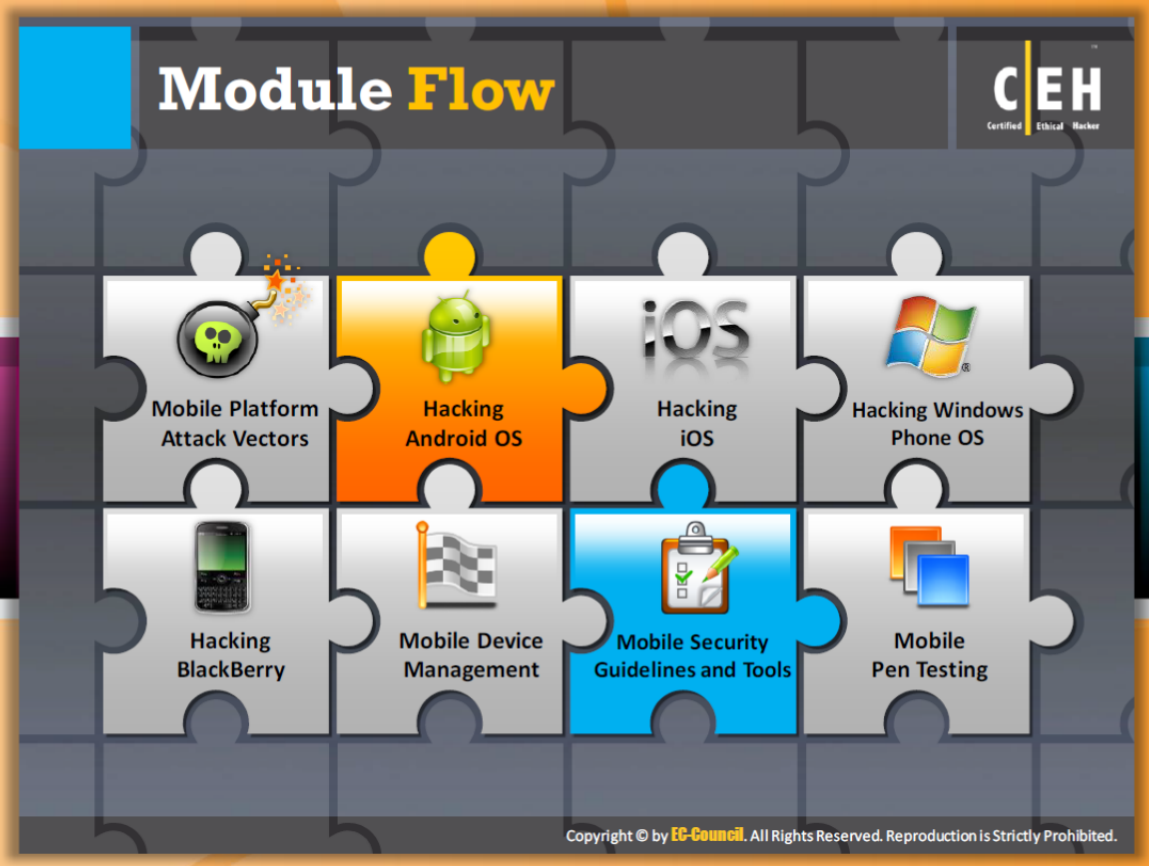


FIGURE 16.6: Vulnerable Sandbox Environment



Module Flow

So far, we have discussed various potential attack vectors of mobile platforms. Now we will discuss hacking the Android OS.

Mobile Platform Attack Vectors	Hacking BlackBerry
Hacking Android iOS	Mobile Device Management
Hacking iOS	Mobile Security Guidelines and Tools
Hacking Windows Phone OS	Mobile Pen Testing

This section introduces you to the Android OS and its architecture, various vulnerabilities associated with it, Android rooting and Android rooting tools, various Android Trojans, Android security tools, Android penetration **testing tools**, and Android device tracking tools.

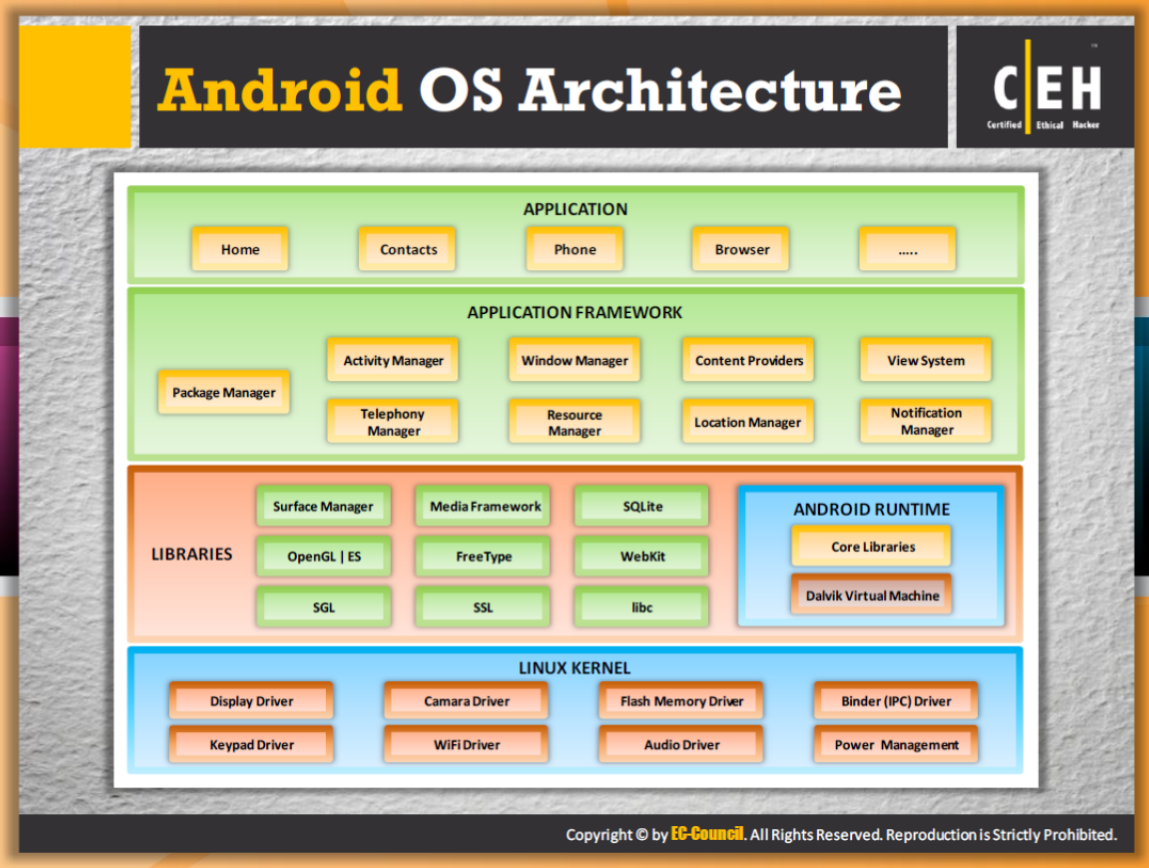
Android OS

Android is a **software stack** developed by Google specifically for mobile devices such as smartphones and tablet computers. It is comprised of an operating system, middleware, and key applications. Android's mobile operating system is based on the **Linux kernel**. The Android application runs in a **sandbox**. The sandbox security mechanism is explained on a previous slide. Antivirus software such as Lookout Mobile Security, AVG Technologies, and McAfee are released by security firms for Android devices. However, the sandbox is also applicable to the antivirus software. As a result, though this **antivirus** software has the ability to scan the complete system, it is limited to scanning up to a certain environment.

The features of android operating system include:

- **Application framework** enabling reuse and replacement of components
- Dalvik virtual machine optimized for mobile devices
- Integrated browser based on the open source WebKit engine
- SQLite for structured data storage
- Media support for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)

- Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE



Android OS Architecture

Android is a **Linux-based operating system** especially designed for **portable devices** such as smartphones, tablets, etc. The pictorial representation that follows shows the different layers such as application, application framework, libraries, android runtime, and Linux kernel, which make up the Android operating system.

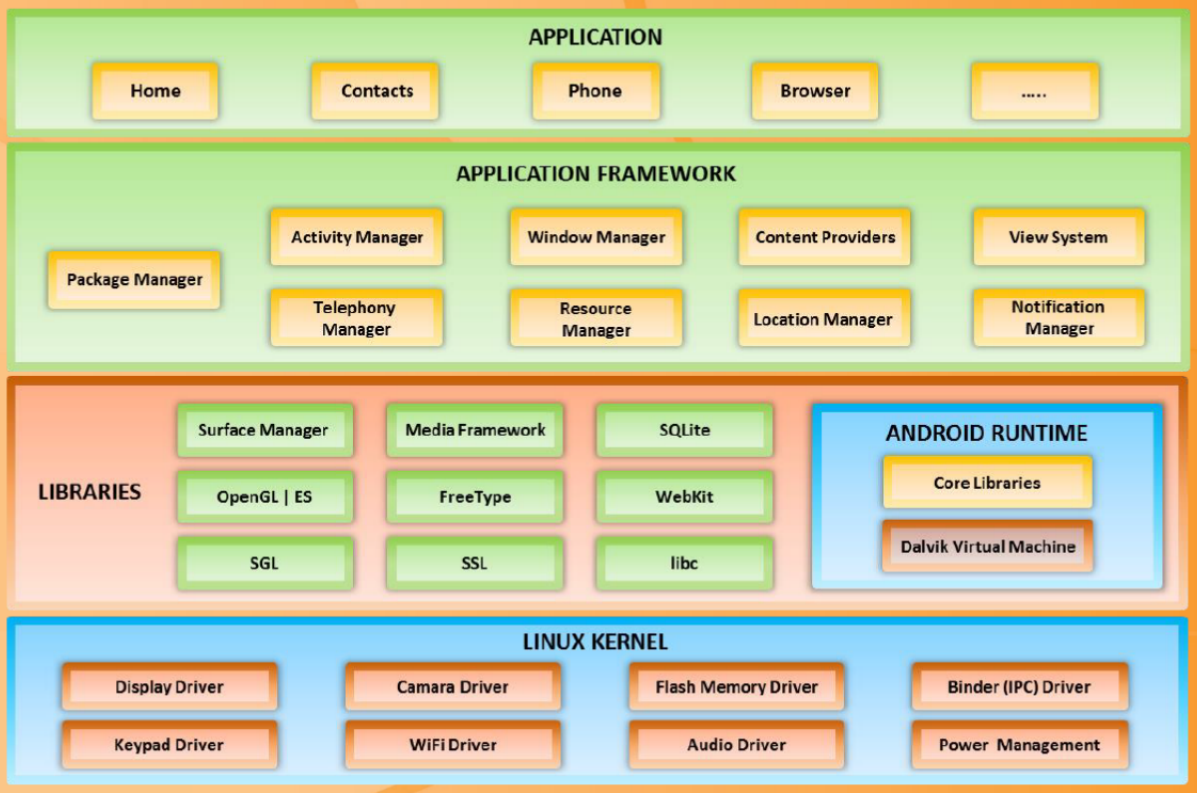


FIGURE 16.7: Android OS Architecture

Applications:

The applications provided by Android include an email client, SMS, calendar, maps, Browser, contacts, etc. These applications are written using the **Java programming language**.

Application Framework

- As Android is an **open development platform**, developers have full access to the API that is used in the core applications
- The View System can be used to develop lists, grids, text boxes, buttons, etc. in the application
- The Content Provider permits applications to access data from other applications in order to share their own data
- The Resource Manager allocates the **non-code resources** like localized strings, graphics, etc.
- The Notification Manager helps applications to show custom messages in the status bar
- The Activity Manager controls the lifecycle of applications

Libraries

Libraries comprise each and every code that provides the main features of an Android OS. For example, database support is provided by the **SQLite library** so that an application can utilize it for storing data and functionalities for the web browser provided by the **Web Kit library**. The

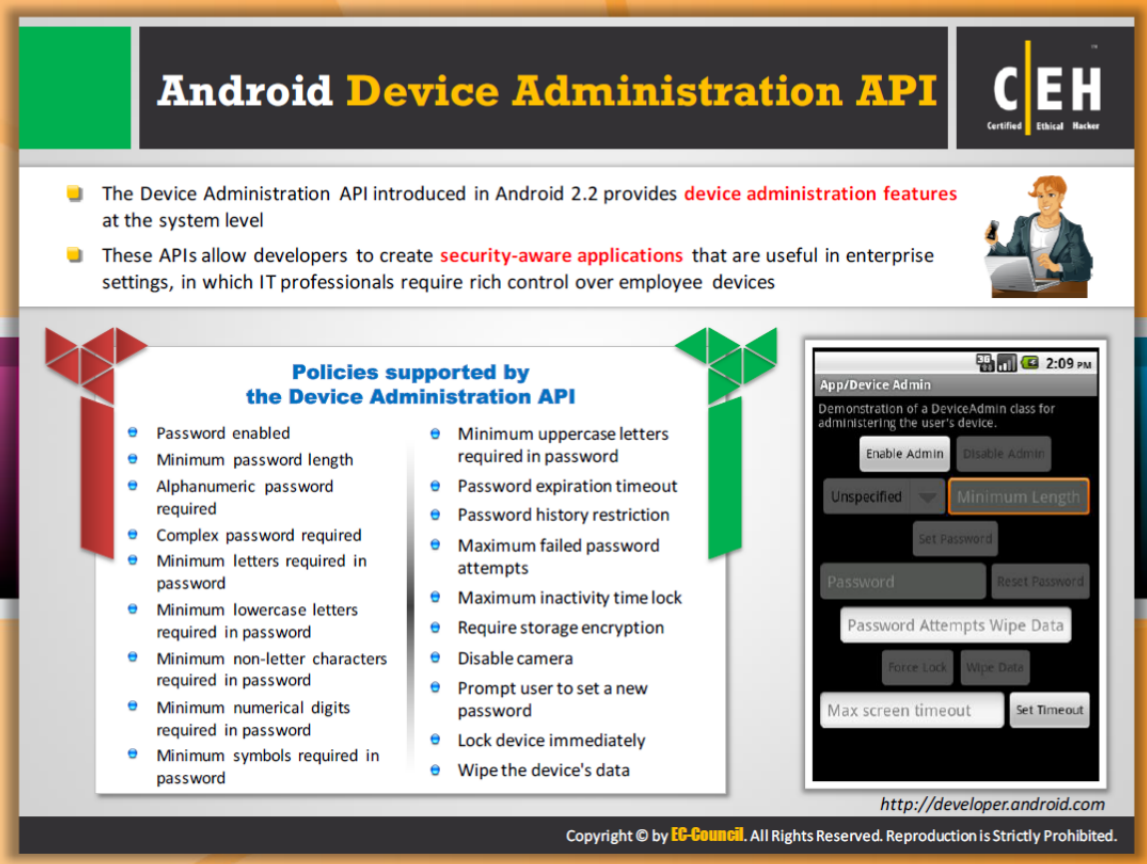
Android core library includes Surface Manager, Media Framework, SQLite, OpenGL | ES, FreeType, WebKit, SGL, SSL, libc, SQLite (database engine), and LibWebCore (web browser engine).

Android Runtime

Android Runtime includes **core libraries** and the **Dalvik virtual machine**. The set of core libraries allows developers to write the Android applications using the Java programming language. Dalvik virtual machine is helpful in executing Android applications. Dalvik can run multiple VMs efficiently.

Linux Kernel

The Android operating system was built based on the **Linux kernel**. This layer is made up of all the low-level device drivers such as Display Driver, Camara Driver, Flash Memory Driver, Binder (IPC) Driver, Keypad Driver, WiFi Driver, Audio Driver, and Power Management for various hardware components of an Android device.



Android Device Administration API

Source: http://developer.android.com

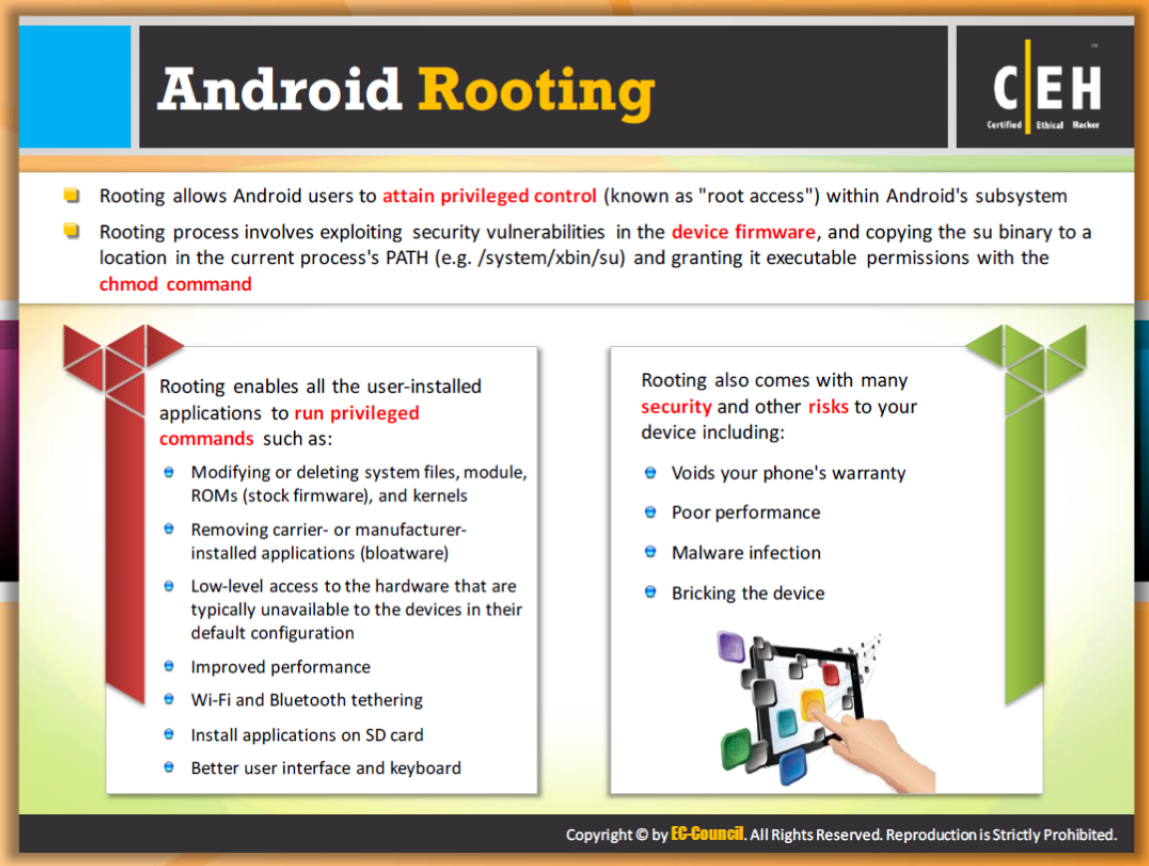
The **Device Administration API** introduced in **Android 2.2** provides device administration features at the system level. These APIs allow developers to create security-aware applications that are useful in enterprise settings, in which IT professionals require rich control over employee devices. The device admin applications are written using the **Device Administration API**. These device admin applications enforce the desired policies when the user installs these applications on his or her device. The **built-in applications** can leverage the new APIs to improve the exchange support.

Policy	Description
Password enabled	Requires that devices ask for PIN or passwords.
Minimum password length	Set the required number of characters for the password. For example, you can require PIN or passwords to have at least six characters.
Alphanumeric password required	Requires that passwords have a combination of letters and numbers. They may include symbolic characters.

Complex password required	Requires that passwords must contain at least a letter, a numerical digit, and a special symbol. Introduced in Android 3.0.
Minimum letters required in password	The minimum number of letters required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum lowercase letters required in password	The minimum number of lowercase letters required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum non-letter characters required in password	The minimum number of non-letter characters required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum numerical digits required in password	The minimum number of numerical digits required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum symbols required in password	The minimum number of symbols required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum uppercase letters required in password	The minimum number of uppercase letters required in the password for all admins or a particular one. Introduced in Android 3.0.
Password expiration timeout	When the password will expire, expressed as a delta in milliseconds from when a device admin sets the expiration timeout. Introduced in Android 3.0.
Password history restriction	This policy prevents users from reusing the last *n* unique passwords. This policy is typically used in conjunction with **setPasswordExpirationTimeout()**, which forces users to update their passwords after a specified amount of time has elapsed. Introduced in Android 3.0.
Maximum failed password attempts	Specifies how many times a user can enter the wrong password before the device wipes its data. The Device Administration API also allows administrators to remotely reset the device to factory defaults. This secures data in case the device is lost or stolen.
Maximum inactivity time lock	Sets the length of time since the user last touched the screen or pressed a button before the device locks the screen. When this happens, users need to enter their PIN or passwords again before they can use their devices and access data. The value can be between 1 and 60 minutes.
Require storage encryption	Specifies that the storage area should be encrypted, if the device supports it. Introduced in Android 3.0.
Disable camera	Specifies that the camera should be disabled. Note that this doesn't have to be a permanent disabling. The camera can be enabled/disabled dynamically based on context, time, and so on. Introduced in Android 4.0.

TABLE16.1: Android Device Administration API

FIGURE 16.8: Android Device Administration API



Android Rooting

Rooting is the process of removing the limitations and allowing full access. It allows Android users to attain “super user” privileged control (known as "**root access**") and permission within Android's **subsystem**. After rooting the Android phone, an Android user will have control over SETTINGS, FEATURES, and PERFORMANCE of his or her phone and can even install software that is not supported by the device. The root users will have “super –user” privileges using which they can easily alter or modify the software code on the device. Rooting is basically hacking Android devices and is equivalent to "**jailbreaking**" in iPhone. Rooting exploits a security vulnerability in the device firmware, and copying the su binary to a location in the current process's PATH (e.g. /system/xbin/su) and granting it executable permissions with the chmod command.

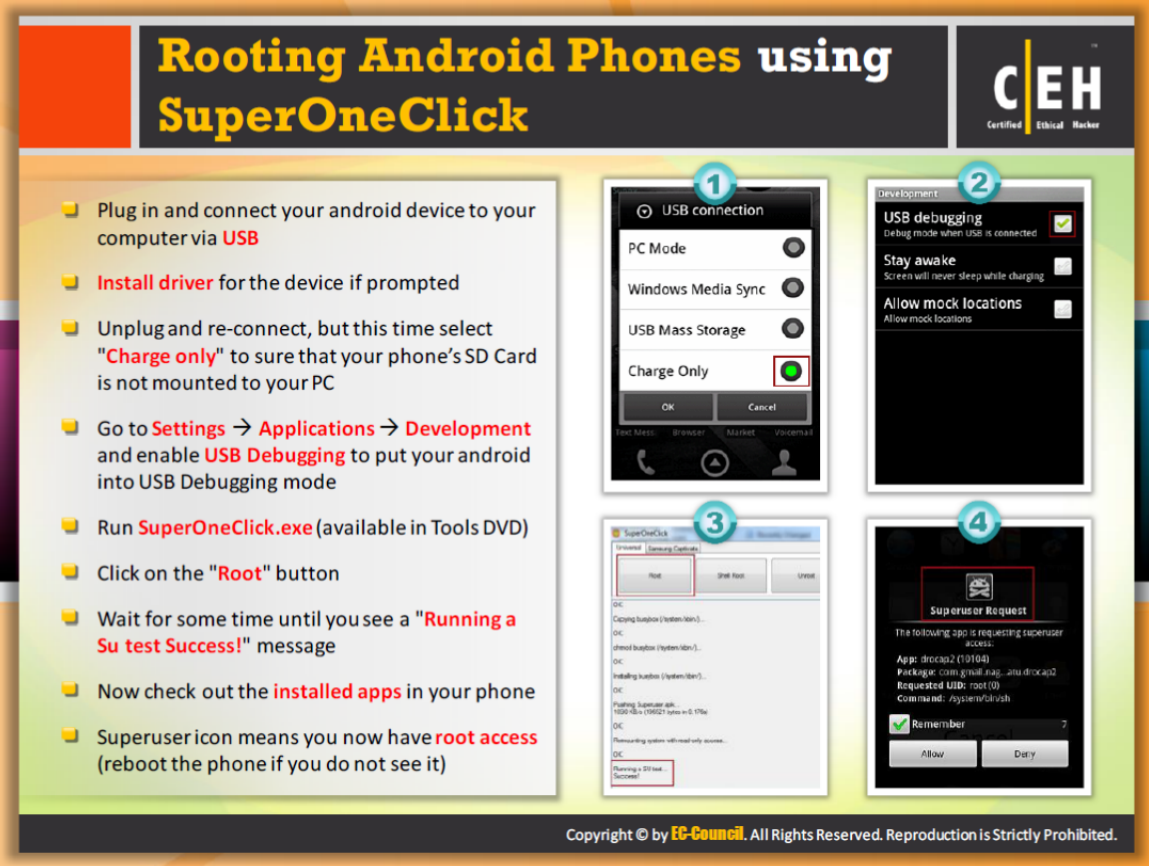
Rooting enables all the user-installed applications to run privileged commands such as:

- Modifying or deleting system files, module, ROMs (stock firmware), and kernels
- Removing carrier- or manufacturer-installed applications (bloatware)
- Low-level access to the hardware that are typically unavailable to the devices in their default configuration
- Improved performance

- Wi-Fi and Bluetooth tethering
- Install applications on SD card
- Better user interface and keyboard

Rooting also comes with many security and other risks to your device including:

- Voids your phone's warranty
- Poor performance
- Malware infection
- Bricking the device



Rooting Android Phones using SuperOneClick

SuperOneClick is a tool designed especially for rooting an Android phone. The step-by-step procedure for rooting an Android phone with the help of **SuperOneClick** follows:

- Plug in and connect your Android device to your computer via a USB.
- Install the driver for the device if prompted.
- Unplug and re-connect, but this time select **Charge only** to ensure that your phone's SD Card is not mounted to your PC.
- Go to **Settings → Applications → Development** and enable USB Debugging to put your android into USB Debugging mode.
- Run SuperOneClick.exe (available in Tools DVD).
- Click the Root button.
- Wait for some time until you see a "Running a Su test Success!" message
- Now check out the installed apps in your phone.
- Superuser icon means you now have root access (reboot the phone if you don't see it).

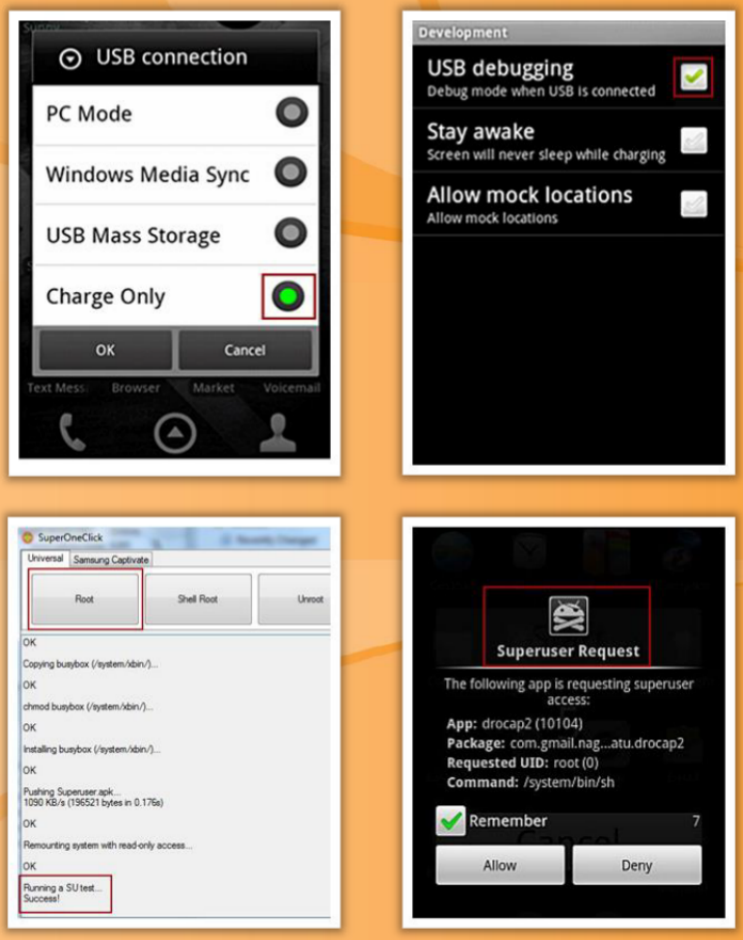
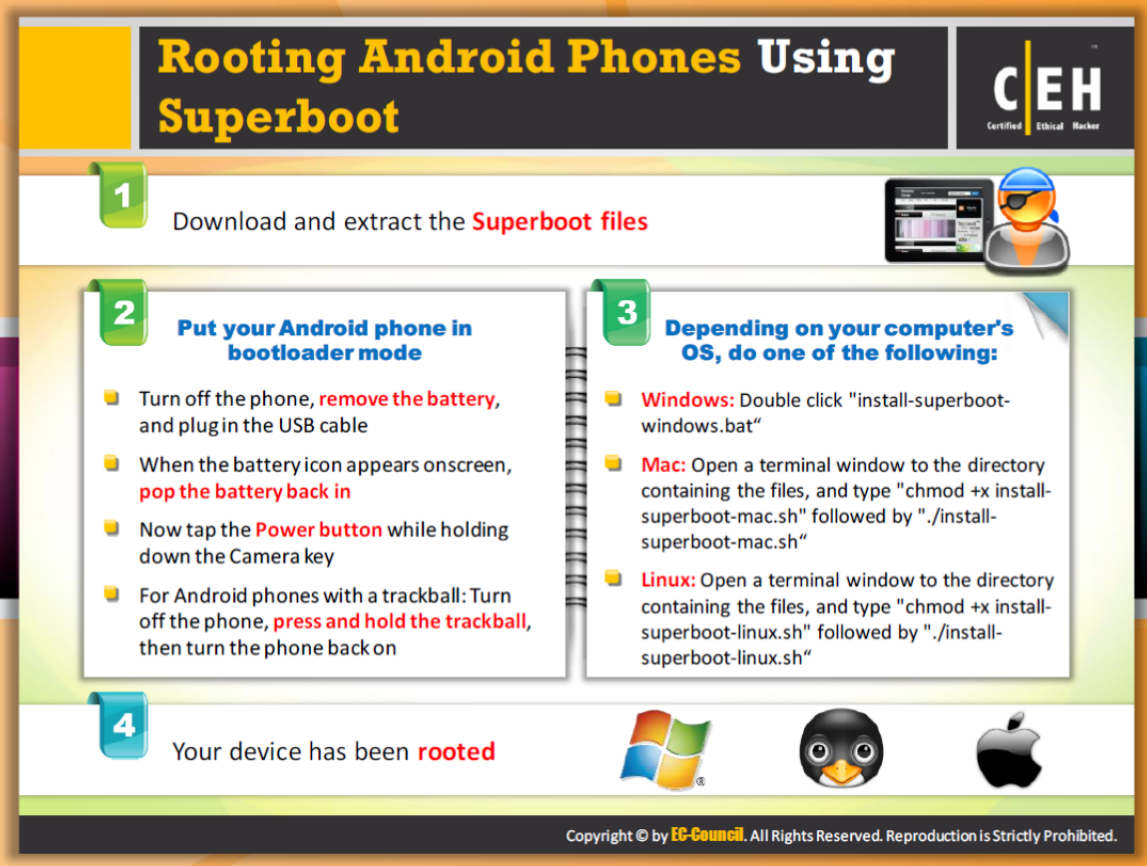


FIGURE 16.9: Rooting Android Phones using SuperOneClick



Rooting Android Phones using Superboot

Superboot is a **boot.img**. It is designed specifically to root Android phones. It roots Android phones when they are booted for the very first time. Any individual can root the Android phone using **superboot** by following these steps:

Step 1: Download and extract the **Superboot files**.

Step 2: Put your Android phone in **bootloader mode**:

- Turn off the phone, remove the battery, and plug in the USB cable.
- When the battery icon appears onscreen, pop the battery back in.
- Now tap the Power button while holding down the Camera key.
- For Android phones with a trackball: Turn off the phone, press and hold the trackball, then turn the phone back on.

Step 3: Depending on your computer's OS, do one of the following:

- **Windows**: Double-click **install-superboot-windows.bat**.
- **Mac**: Open a terminal window to the directory containing the files, and type chmod +x install-superboot-mac.sh" followed by ./install-superboot-mac.sh.

- **Linux**: Open a terminal window to the directory containing the files, and type chmod +x install-superboot-linux.sh" followed by ./install-superboot-linux.sh.

Step 4: Your Android device has been rooted.

Android Rooting Tools

In addition to **SuperOneClick** and **Superboot**, there are many other tools that can be used for rooting Android phones:

- Unrevoked available at http://unrevoked.com
- Recovery Flasher available at https://sites.google.com/site/adlxmod
- Universal Androot available at http://forum.xda-developers.com
- Unlock Root available at www.unlockroot.com

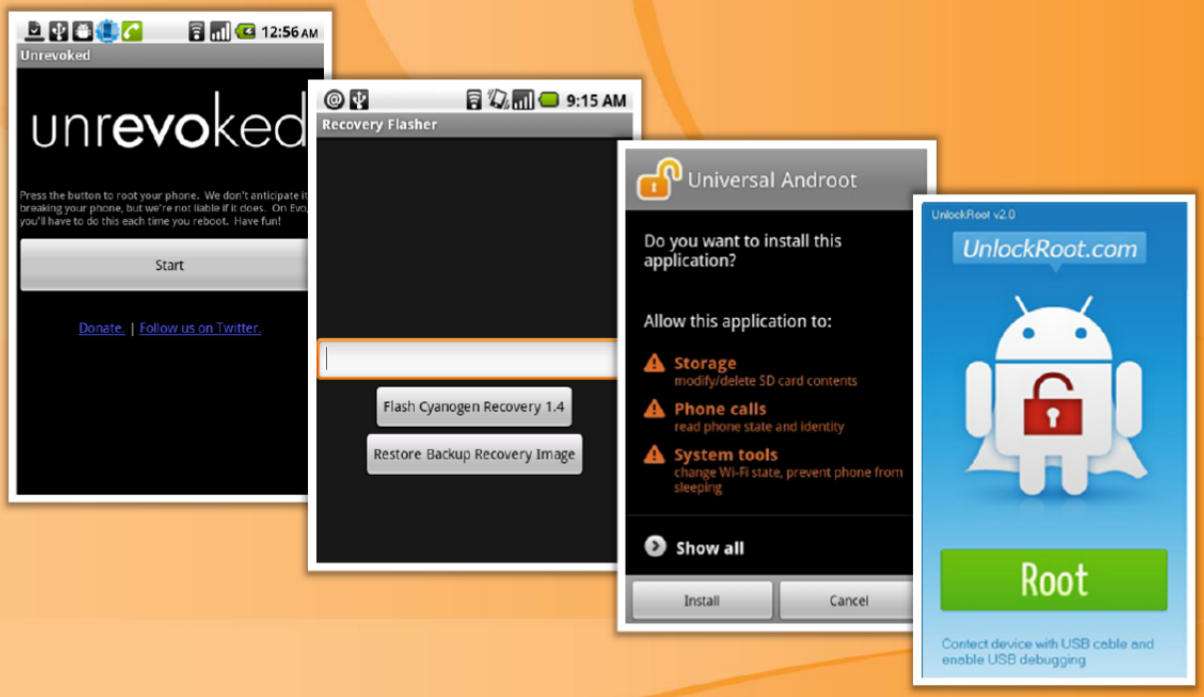
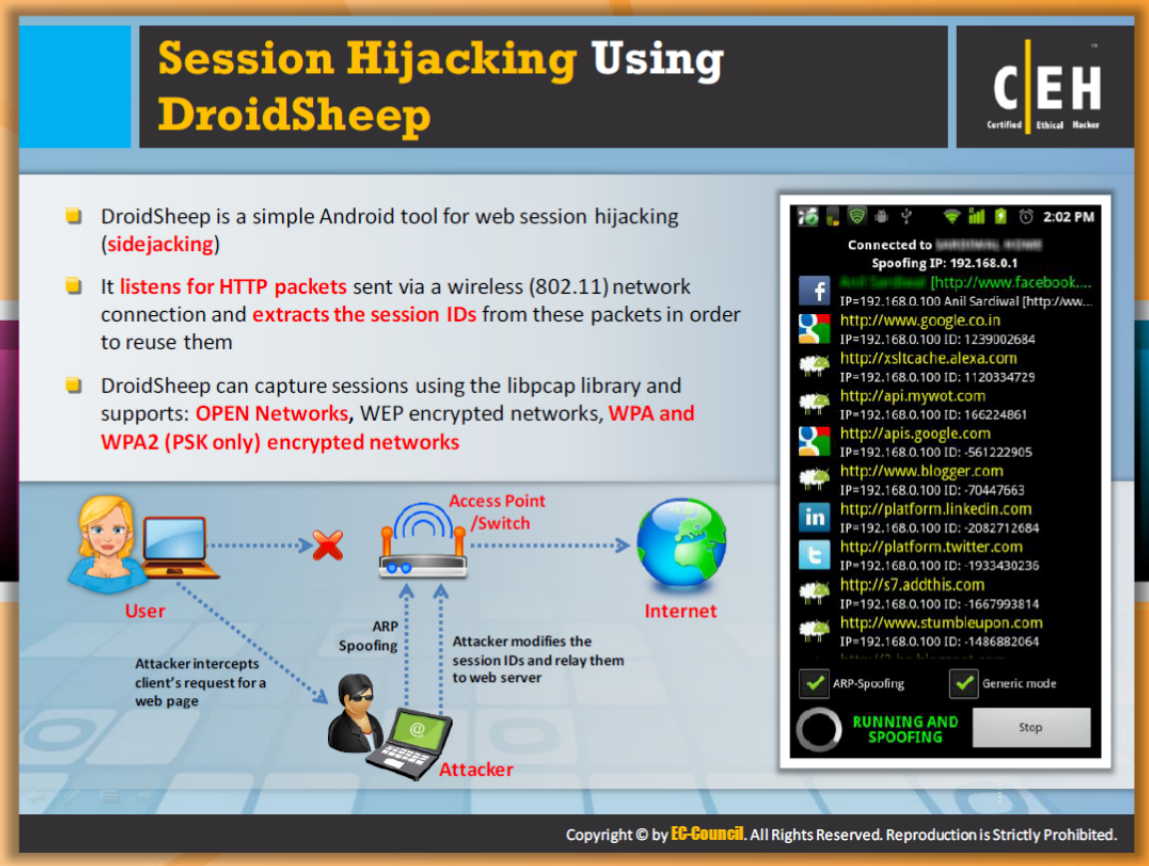


FIGURE 16.10: Android Rooting Tools



Session Hijacking Using DroidSheep

Most web applications use a **session ID** to verify the user's identity with the application. This session ID is transmitted in subsequent requests within HTTP packets in order to maintain the session with the user. The attacker uses the **DroidSheep tool** to read the all the packets sent via a wireless network and captures the session ID. Once the attacker captures the victim's legitimate session ID, he or she may use this stolen session ID to access the target web application on behalf of the victim.

DriopSheep **listens and captures** HTTP packets sent via a wireless (**802.11**) network and then analyzes the captured packets to extract and reuse the session IDs. DriopSheep accomplishes this using the libcap library. It supports OPEN Networks, WEP encrypted networks, WPA, and WPA2 (PSK only) encrypted networks.

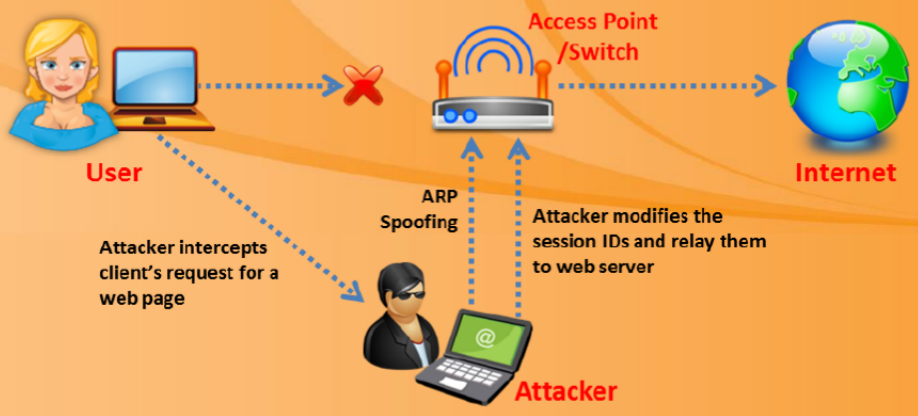


FIGURE 16.11: Session Hijacking Using DroidSheep

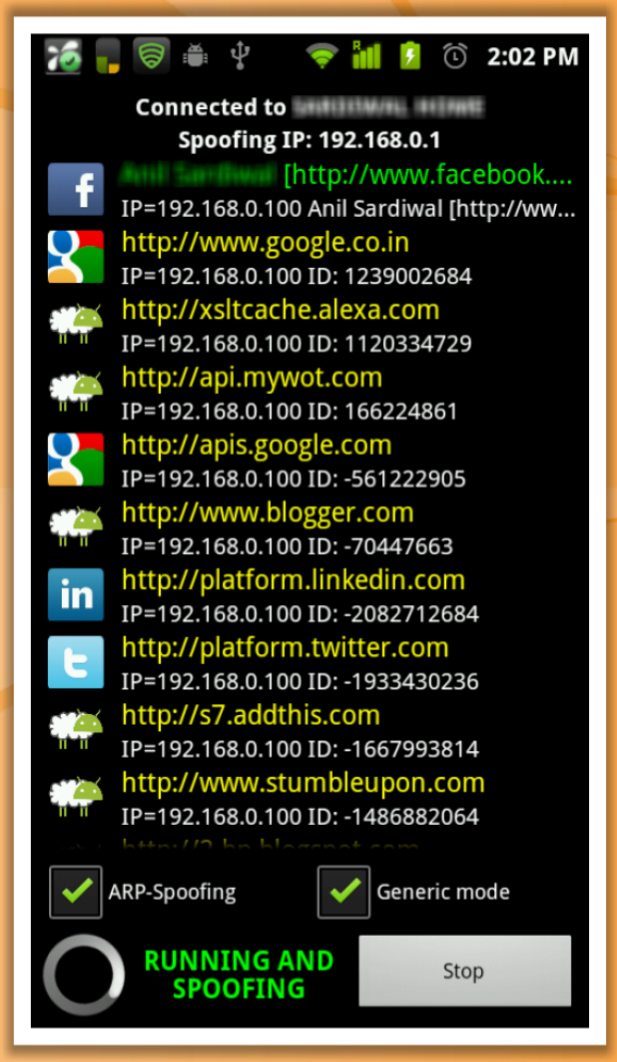


FIGURE 16.12: DroidSheep Screenshot

Android-based Sniffer: FaceNiff

Source: http://faceniff.ponury.net

FaceNiff is an **Android app** that allows you to sniff and intercept web session profiles over the Wi-Fi that your mobile is connected to. It is possible to hijack sessions only when Wi-Fi is not using EAP, but it should work over any private networks (**Open/WEP/WPA-PSK/WPA2-PSK**).

Note: If webuser uses SSL this application won't work.

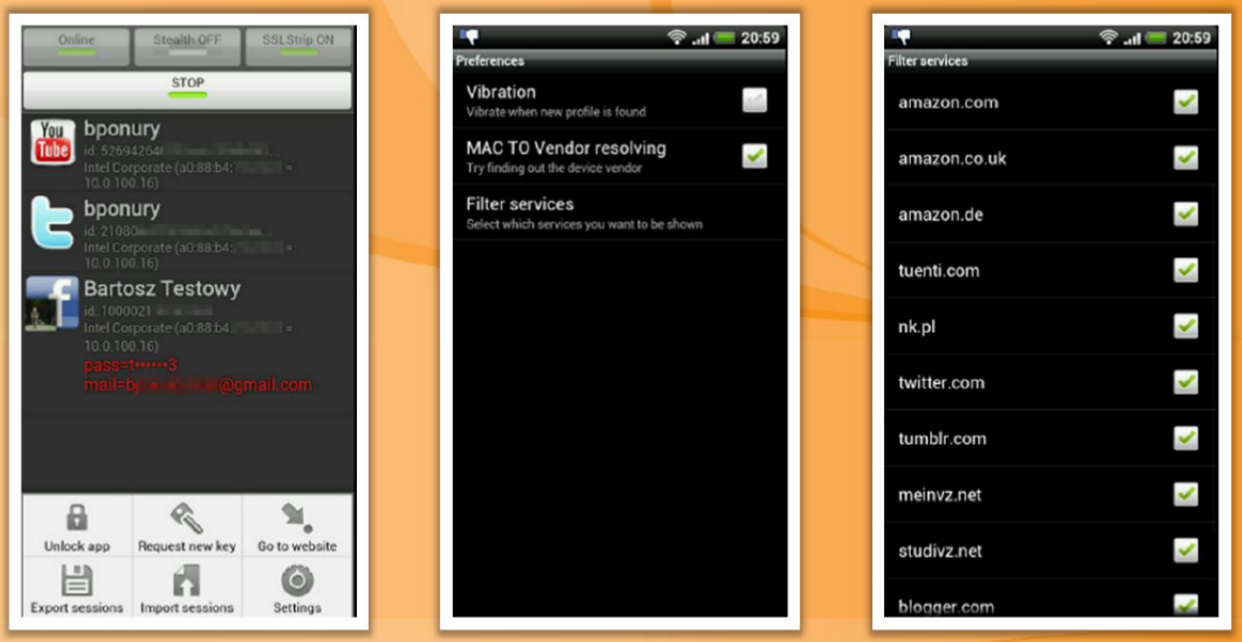
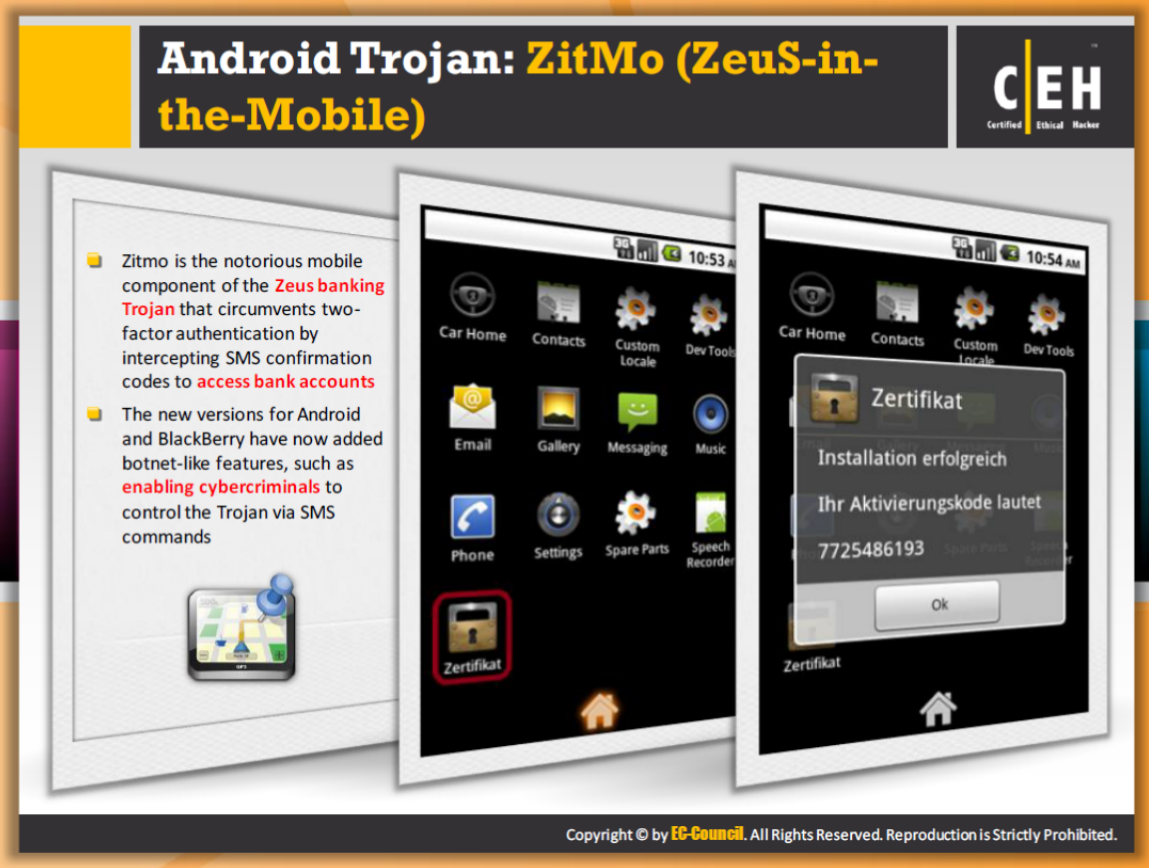


FIGURE 16.13: FaceNiff Screenshot



Android Trojan: ZitMo (ZeuS-in-the-Mobile)

Zitmo refers to a version of the Zeus malware that specifically **targets mobile devices**. It is a malware Trojan horse designed mainly to steal online banking details from users. It circumvents mobile banking app security by simply forwarding the infected mobile's SMS messages to a command and control mobile owned by cybercriminals. The new versions of Android and BlackBerry have now added botnet-like features, such as enabling **cybercriminals** to control the Trojan via **SMS commands**.

FIGURE 16.14: ZitMo (ZeuS-in-the-Mobile) Screenshot

Android Trojan: GingerBreak

AndroidOS/GingerBreak is a **Trojan** that affects mobile devices running the Android operating system. It drops and executes another Trojan detected as Exploit: **AndroidOS/CVE-2011-1823**, which, if run successfully, gains administrator privileges on the device.

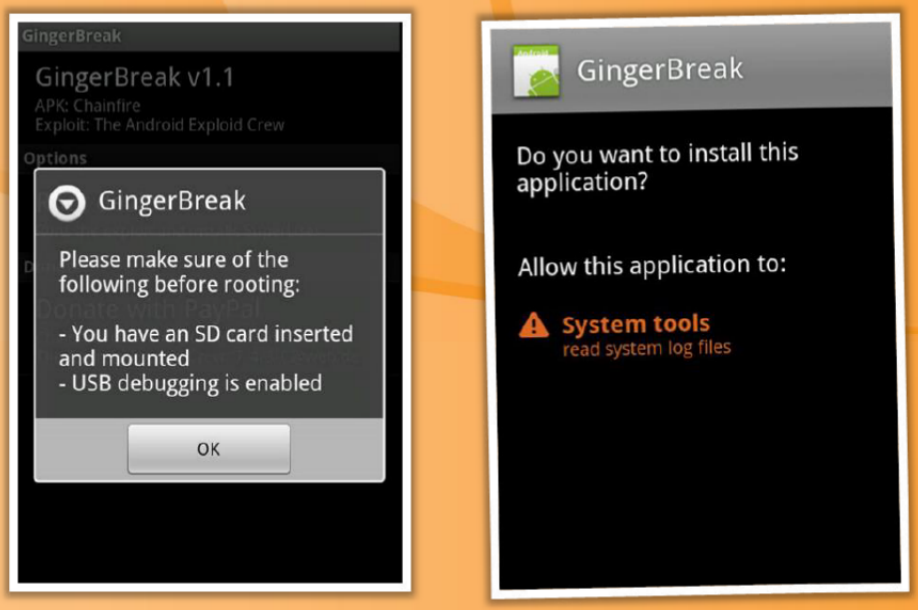
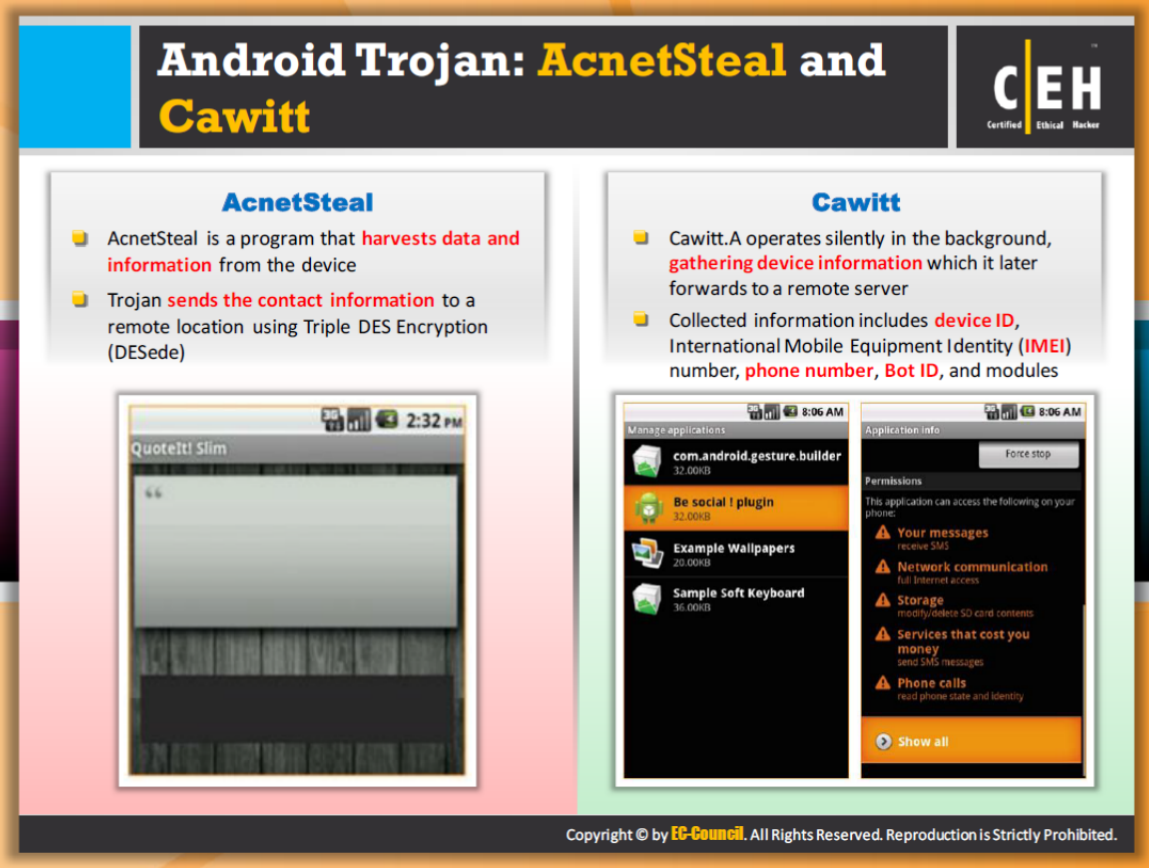


FIGURE 16.15: GingerBreak Screenshot



Android Trojan: AcnetSteal and Cawitt

AcnetSteal

AcnetSteal is a program that **harvests data** and information from the device. The Trojan sends the contact information to a remote location using Triple **DES Encryption (DESede)**.

FIGURE 16.16: AcnetSteal Screenshot

Cawitt

Cawitt operates silently in the background, gathering device information which it later forwards to a **remote server**. Collected information includes device ID, **International Mobile Equipment Identity (IMEI)** number, phone number, Bot ID, and modules. This Trojan doesn't place any launcher icon in the application menu in order to avoid being detected by the device user.

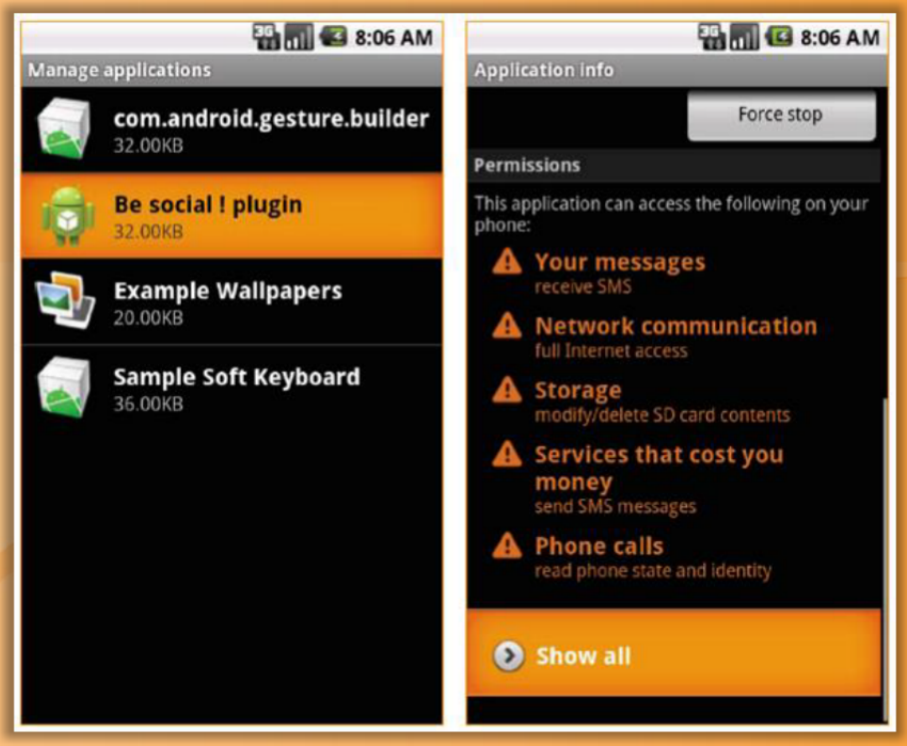
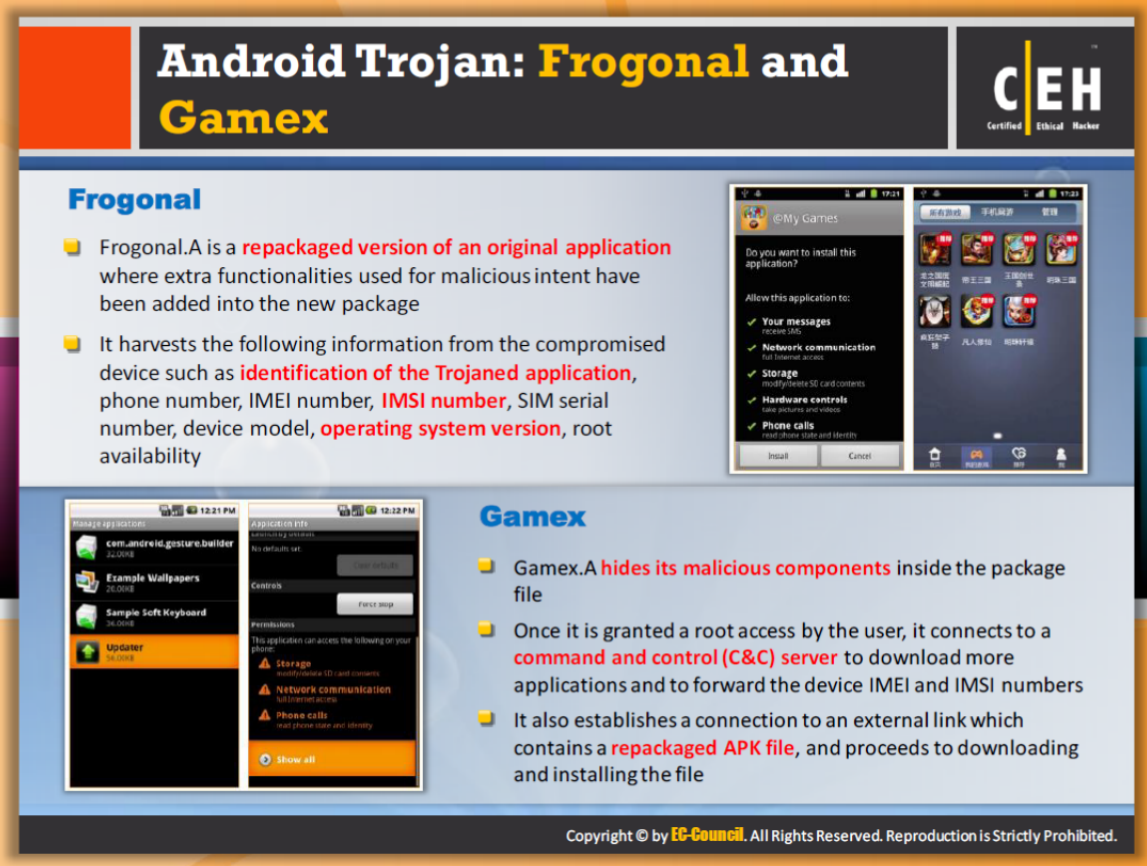


FIGURE 16.17: Cawitt Screenshot



Android Trojan: Frogonal and Gamex

Frogonal

Frogonal is a **repackaged version** of an original application where extra functionalities used for malicious intent have been added into the new package. It harvests the following information from the compromised mobile devices:

- Identification of the **Trojanized** application:
  - Package name
  - Version code
- Phone number
- IMEI number
- IMSI number
- SIM serial number
- Device model
- Operating system version
- Root availability

FIGURE 16.18: Frogonal and Gamex Frogonal Screenshot

Gamex

Gamex is an **Android Trojan** that downloads and installs the files on a compromised mobile device. It hides the malicious content inside the file that is to be installed; once it is granted a root access by the device owner, it connects to a **command and control (C&C)** server to download more applications and to forward the device's **IMEI and IMSI** numbers. It also establishes a connection to an external link that contains a repackaged APK file, and proceeds to download and install the file.

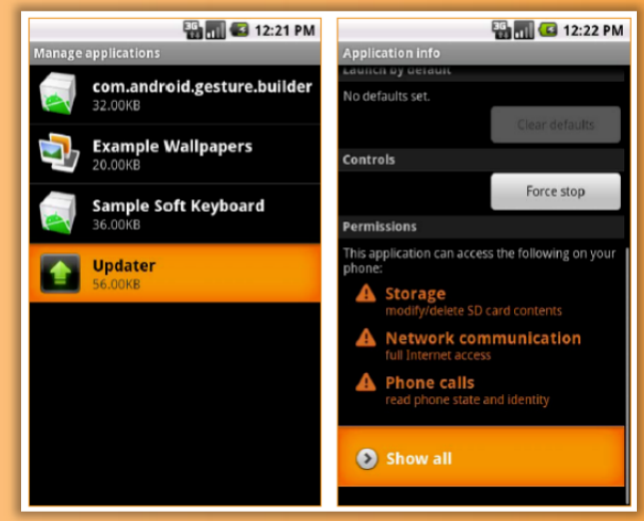
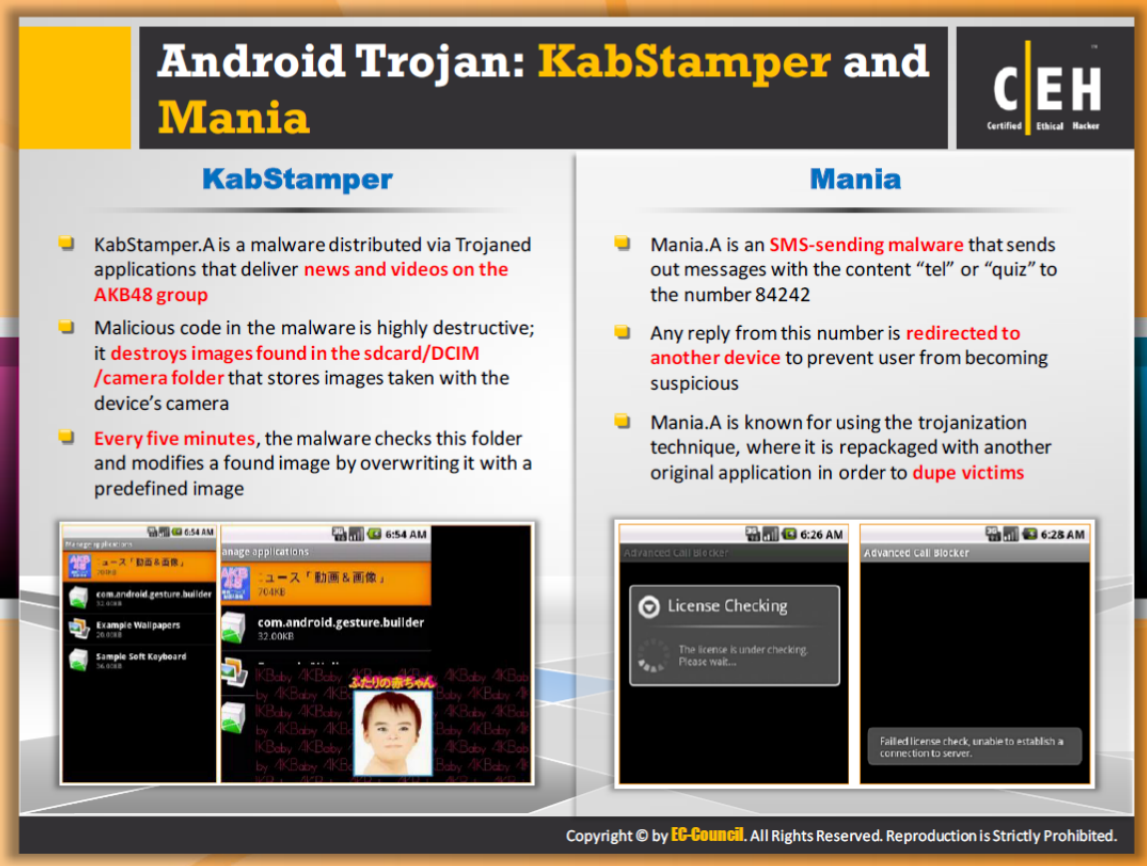


FIGURE 16.19: Gamex Screenshot



Android Trojan: KabStamper and Mania

KabStamper

KabStamper is an Android Trojan that modifies images found in the target mobile device by overwriting them with a predefined image. It is distributed via **Trojanized applications** that deliver news and videos about **the AKB48** group. It is very destructive and destroys images found in the **sdcard/DCIM/camera** folder that stores images taken with the device's camera.

FIGURE 16.20: KabStamper and Mania Kabstamper Screenshot

Mania

Mania is an Android Trojan that pretends to perform license checking to cover up its SMS-sending activities in the background. It is **SMS-sending malware** that sends out messages with the content "tel" or "quiz" to the number **84242**. Any reply from this number is redirected to another device to prevent the device owner from becoming suspicious. While running, **Mania** appears to be performing license checking, but this process always fails and never seems to be completed. The license checking is a coverup for the SMS sending activities that are taking place in the background.

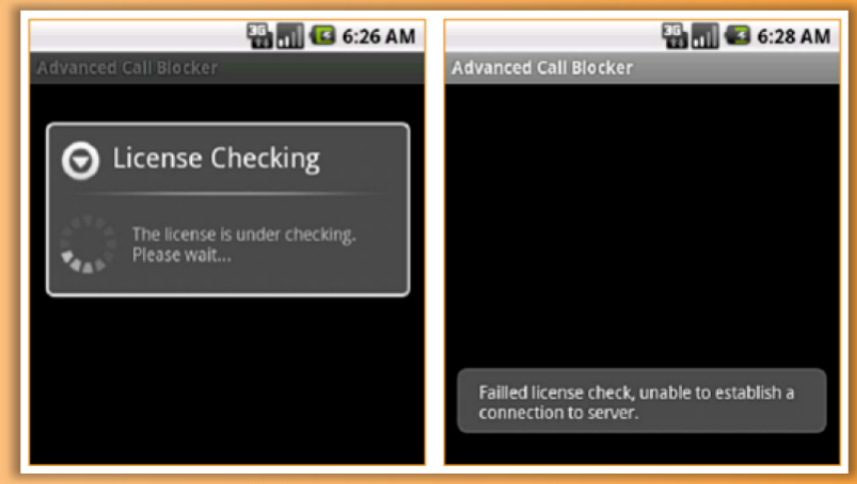
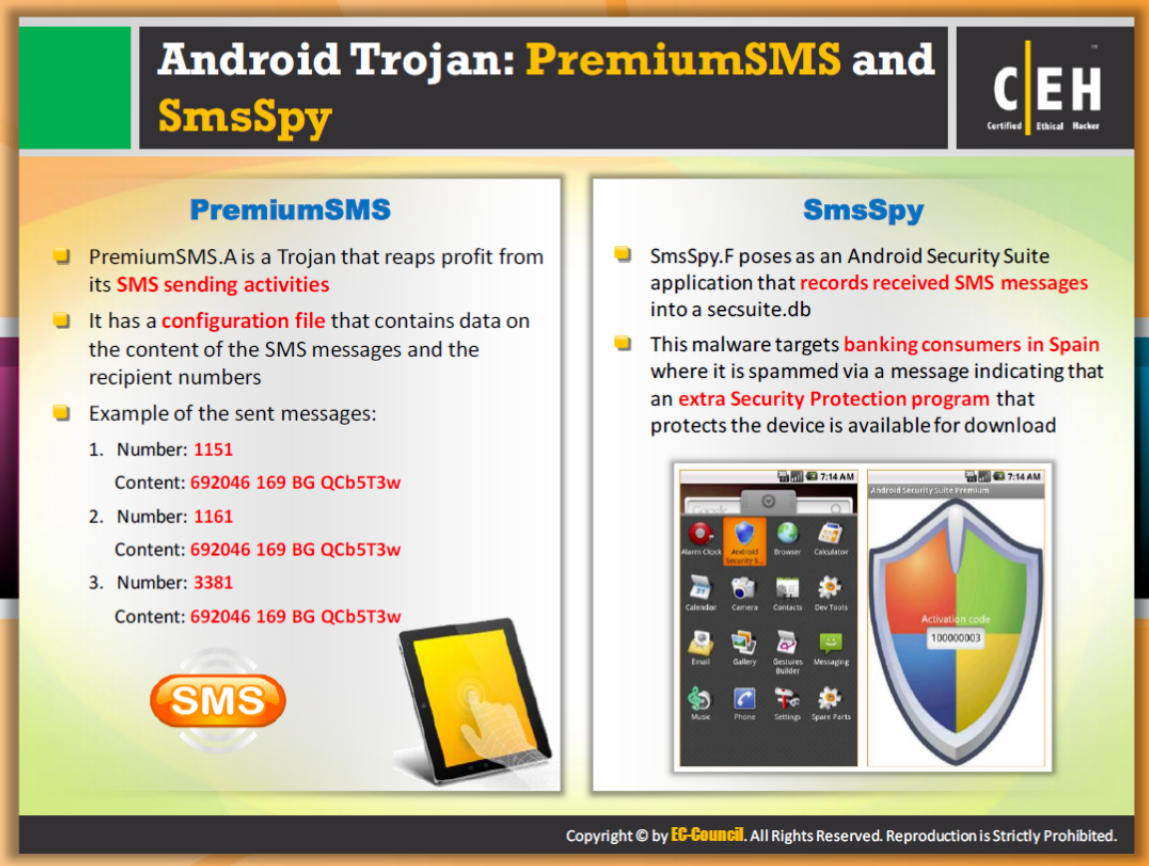


FIGURE 16.21: Mania Screenshot



Android Trojan: PremiumSMS and SmsSpy

PremiumSMS

PremiumSMS is an Android Trojan that reaps profit from its **SMS-sending activities**. It has a configuration file that **contains data** on the content of the SMS messages and the recipient numbers.

Example of send messages:

1. Number: 1151

   Content: 692046 169 BG QCb5T3w
2. Number: 1161

   Content: 692046 169 BG QCb5T3w
3. Number: 3381

   Content: 692046 169 BG QCb5T3w
4. Number: 1005

   Content: kutkut clsamg 6758150
5. Number: 5373

Content: kutkut clsamg 6758150

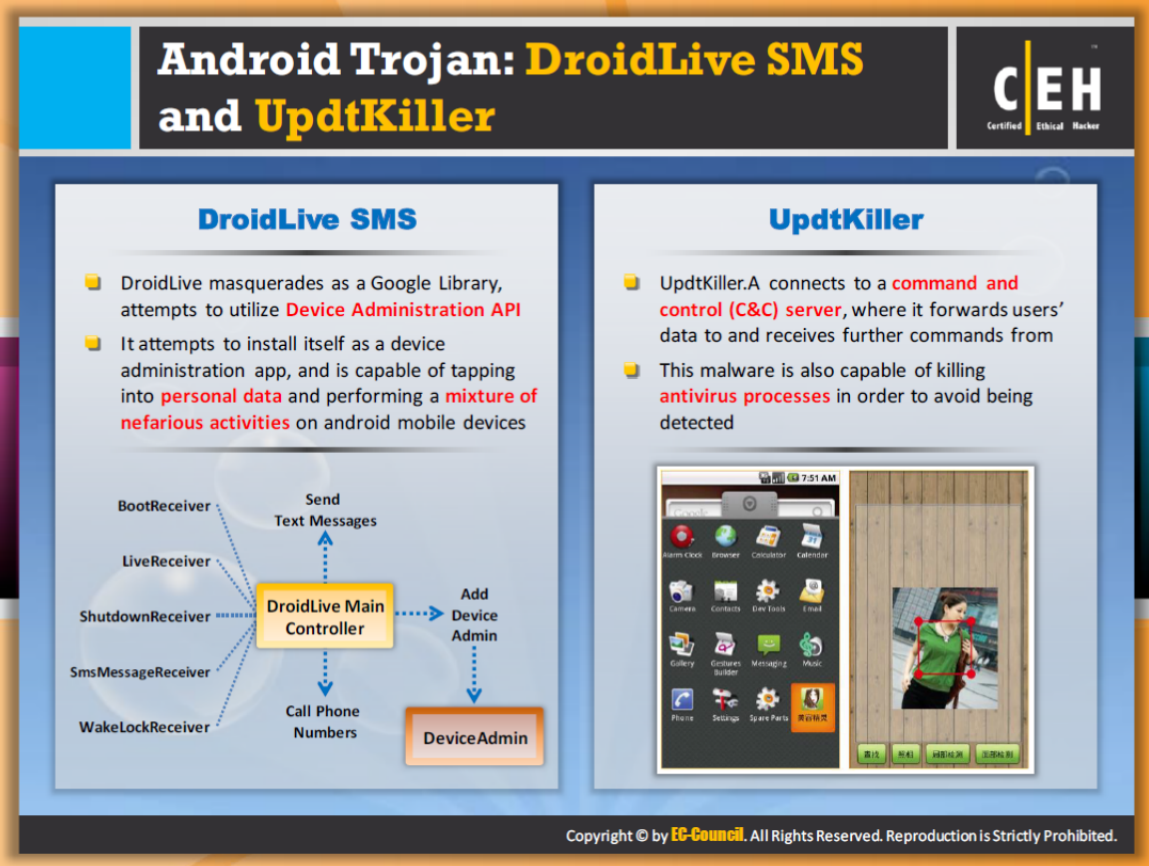
6. Number: 7250

   Content: kutkut clsamg 6758150

SmsSpy

SmsSpy is an Android Trojan that poses as an **Android Security Suite** application that actually does nothing in ensuring the device's security. However, it records received SMS messages into secsuite.db instead. It targets banking consumers in Spain, posing as an Android Security Suite application.

FIGURE 16.22: SmsSpy Screenshot



Android Trojan: DroidLive SMS and UpdtKiller

DroidLive SMS

DroidLive SMS is an Android Trojan masquerading as a **Google Library**; it attempts to utilize a device **administration API**. It attempts to install itself as a device administration app, and is capable of tapping into personal data and performing a mixture of nefarious activities on Android mobile devices. It attempts to disguise itself as a Google library, and receives commands from a **Command and Control (C&C)** server, allowing it to perform functions including sending text messages to premium numbers, initiating phone calls, and collecting personal data.

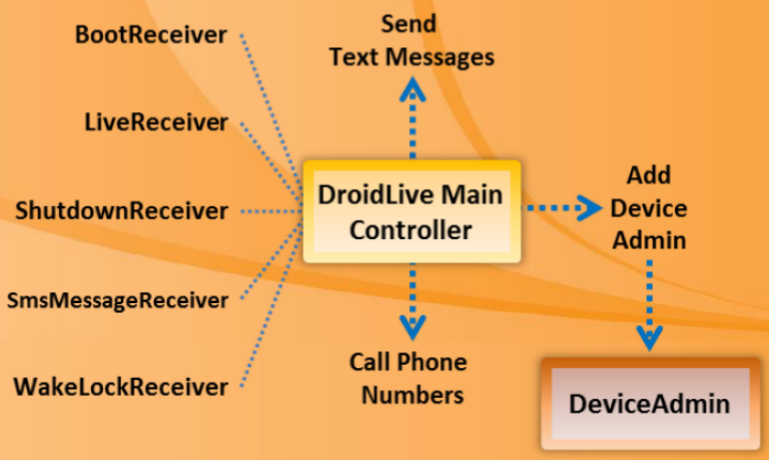


FIGURE 16.23: DroidLive SMS and UpdtKiller DroidLive SMS

Android Trojan: UpdtKiller

UpdtKiller is an Android Trojan that **terminates** processes belonging to antivirus products in order to avoid detection. It connects to a command and control (C&C) server, where it forwards **harvested** user data to and receives further command from.

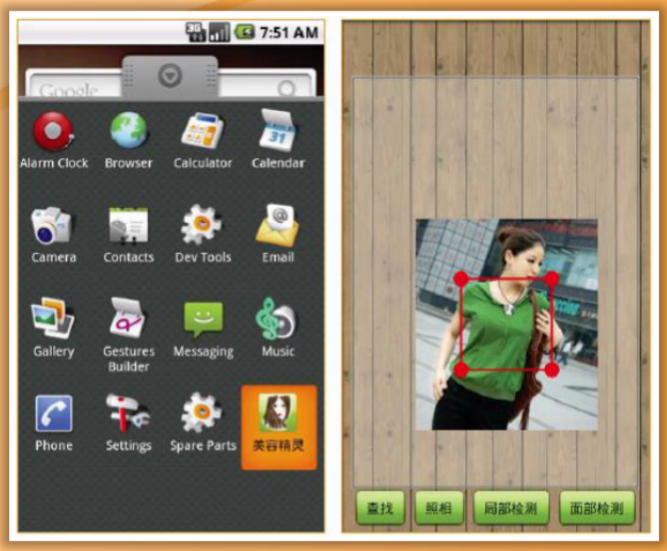


FIGURE 16.24: UpdtKiller Screenshot

Android Trojan: FakeToken

FakeToken steals both authentication factors (**Internet password** and **mTAN**) directly from the mobile device.

Distribution Techniques:

- Through phishing emails **pretending** to be sent by the targeted bank
- Injecting web pages from infected computers, simulating a fake security app that presumably avoids the interception of **SMS messages** by generating a unique digital certificate based on the phone number of the device
- Injecting a phishing web page that redirects users to a website pretending to be a security vendor that offers the "**eBanking SMS Guard**" as protection against "SMS message interception and mobile Phone SIM card cloning"

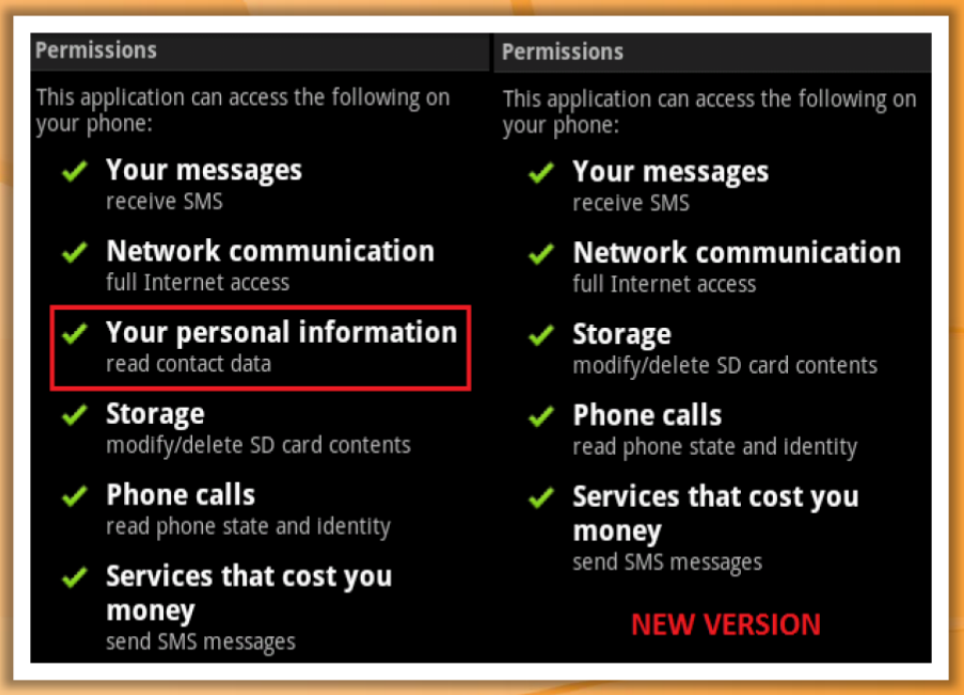
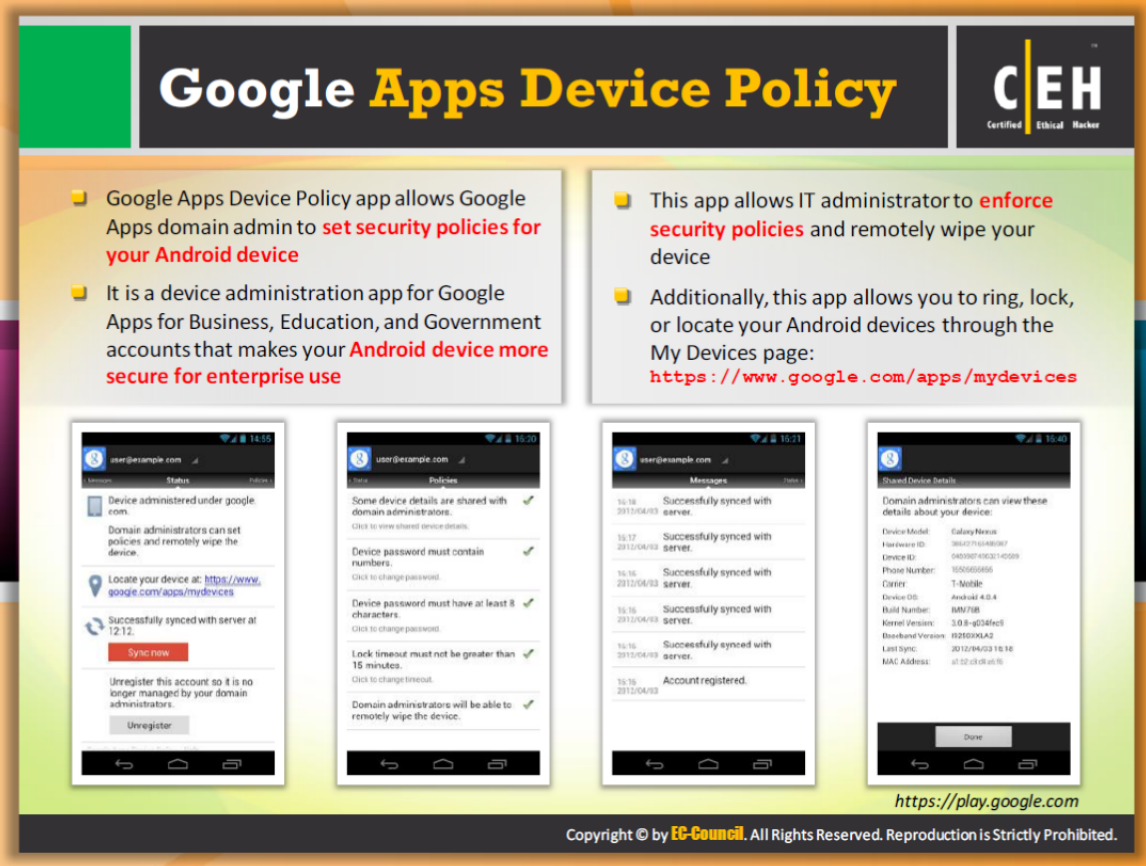


FIGURE 16.25: FakeToken Screenshot

Securing Android Devices

Security of Android devices is a major concern as most people at present using these devices as substitutes for computers. Similar to a traditional computer, security is **mandatory** for Android devices to avoid being **infected** by a malicious application or data loss. The following are a few key points that help you in securing your Android device:

- Enable screen locks for your Android phone for it to be more secure
- Never root your **Android device**
- Download apps only from official Android market
- Keep your device updated with Google Android antivirus software
- Do not directly download **Android package files (APK)**
- Keep updated with the operating system as and when updates arrive
- Use free protectors Android apps such as Android Protector. Where you can assign passwords to text messages, mail accounts, etc.
- Customize your locked home **screen** with the **user's information**



Google Apps Device Policy

Source: https://play.google.com

The Google Apps **Device Policy** app allows a Google Apps domain admin to set security policies for your Android device. It is a device administration app for Google Apps for Business, Education, and **Government** accounts that makes your Android device more secure for enterprise use. This app allows an IT administrator to enforce **security policies** and remotely wipe your device. Additionally, this app allows you to ring, lock, or locate your **Android devices** through the My Devices page: https://www.google.com/apps/mydevices.

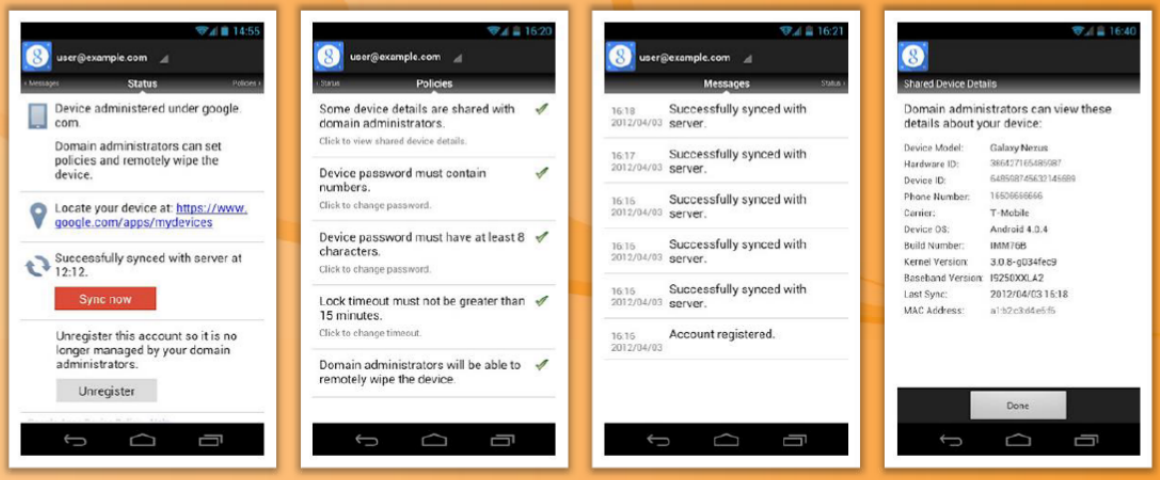


FIGURE 16.26: Google Apps Device Policy

Remote Wipe Service: Remote Wipe

Source: http://support.google.com

Remote Wipe Service is a feature service that allows you to reset or erase the information in the lost or stolen device. To use this service the device should install **Google Sync** or Device Policy. This can also delete all the information in the device such as mail, calendar, and contacts, etc. and cannot delete data stored on the device's SD card. When this service completes its task, it prompts the user with a message as **acknowledgement** to the delete function.

To remote wipe a lost or stolen device:

1. Sign in to your Google Apps control panel.
2. Click **Settings → Mobile**.
3. On the **Devices** tab, hover your cursor over the user whose **device** you want to wipe.
4. Click **Remote Wipe** in the box that appears.
5. A second box appears asking you to confirm that you want to remotely wipe the device. If you are sure you want to wipe the device, click **Wipe Device**.

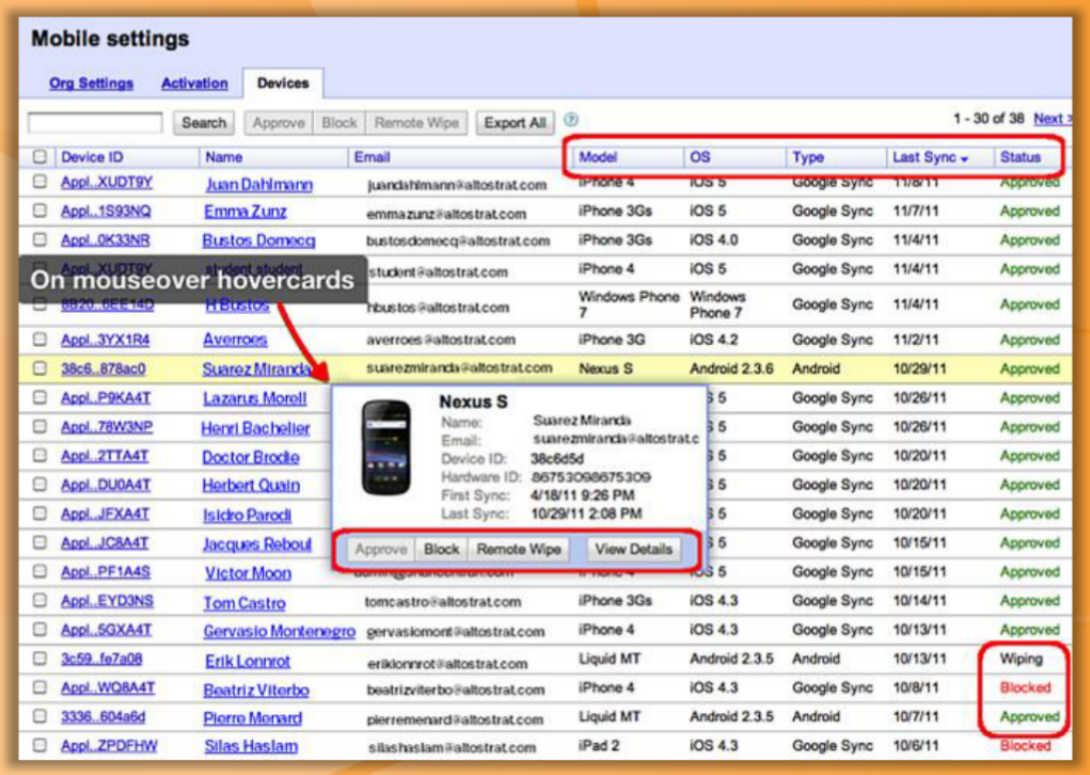


FIGURE 16.27: Remote Wipe Service

Android Security Tool: DroidSheep Guard

Source: http://droidsheep.de

DroidSheep Guard monitors your phone's **ARP-Table** and it warns you by pop-up alerts in case it detects malicious entries. It can instantly disable a Wi-Fi connection to protect your accounts. This can guard against all ARP-based attacks, such as DroidSheep and Faceniff, man-in-middle attacks, handmade attacks, etc. You can use Facebook, eBay, Twitter, and LinkedIn accounts on public Wi-Fis securely.

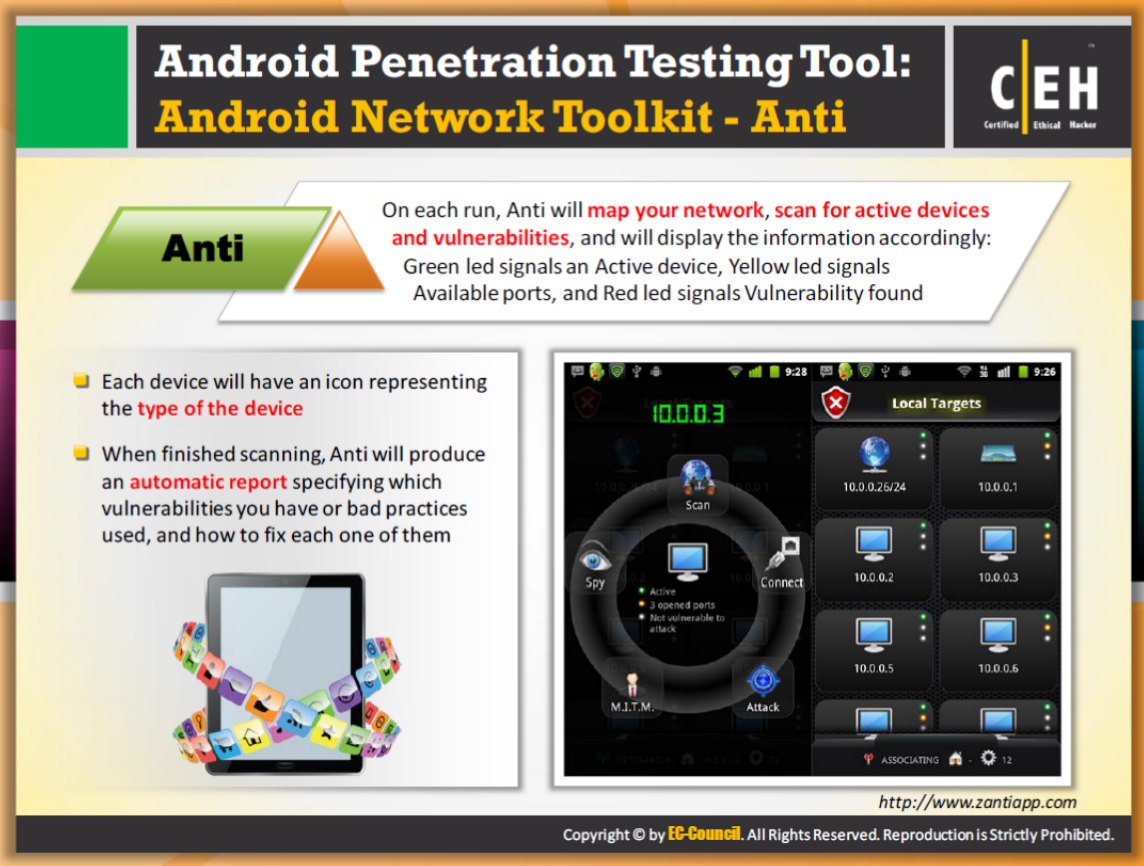
FIGURE 16.28: DroidSheep Guard Screenshot

Android Vulnerability Scanner: X-Ray

Source: http://www.xray.io

X-Ray scans your Android device to determine if there are vulnerabilities that remain unpatched by your carrier. It presents you with a list of vulnerabilities that it is able to identify and allows you to check for the occurrence of vulnerabilities on your device. This is **automatically updated** with the ability to scan for **new vulnerabilities** as they are discovered and disclosed. X-Ray has detailed information about a class of vulnerabilities known as "privilege escalation" vulnerabilities. Such vulnerabilities can be exploited by a malicious application to gain root privileges on a device and perform actions that would normally be restricted by the **Android operating system**.

FIGURE 16.29: X-Ray Screenshot



Android Penetration Testing Tool: Android Network Toolkit - Anti

Source: http://www.zantiapp.com

Android Network Toolkit - Anti is an **Android penetration testing tool**. It is a network scanner that allows you to scan for active devices and vulnerabilities and shows the evidence accordingly: Green signals an "**Active device**," **yellow signals** "**available ports**," and **red signals** "**Vulnerability found**.. Each device has an icon representing the type of device. When finished scanning, it produces an automatic report specifying which vulnerabilities you have or bad practices are used, and how to fix each one of them.

FIGURE 16.30: Android Network Toolkit - Anti

Android Device Tracking Tools

Android device tracking tools help you to track and find the locations of an Android device in case it is lost, stolen, or **misplaced cases**. A few Android device tracking tools are listed as follows:

Find My Phone

Source: http://findmyphone.mangobird.com

Find My Phone is an Android phone app that helps you find your lost, stolen, or misplaced phone. When you lose your phone, just send it a **text msg (SMS)** and the phone will reply with its current location. You can also make your phone ring loudly if you lose it somewhere close, like inside your home.

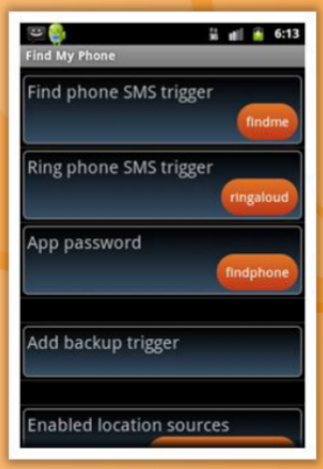


FIGURE 16.31: Find My Phone Screenshot

Prey Anti-Theft

Source: http://preyproject.com

Prey lets you keep track of your laptop, phone, or tablet if it is stolen or missing. It supports **geolocation**. It's lightweight, open source software that gives you full and remote control, 24/7.

FIGURE 16.32: Prey Anti-Theft Screenshot

Android Anti-Theft Security

Source: http://www.snuko.com

The Android anti-theft security tool **Snuko** is anti-theft software that allows you to use it on multiple platforms protecting thousands of PCs, mobile phones, laptops, etc. It offers a complete online back-up solution; as part of the anti-theft package Snuko subscribers' files can be stored safely and securely in the cloud. This can generate important tracking information and security for your data by using its **Mobile Dashboard**. If the mobile device is lost, then the device is locked to prevent any unauthorized access. If the device's SIM card is replaced without

your knowledge, the new SIM card number, phone number, and the **IMEI/IMSI numbers** will be recorded. The phone cannot be used until the correct PIN code is entered.

FIGURE 16.33: Android Anti-Theft Security Screenshot

Wheres My Droid

Source: http://wheresmydroid.com

Where's My Droid is an Android device tracking tool that allows you to **track your phone** from anywhere, either with a text messaged attention word or with an online Commander. The app can also get the GPS coordinates with a link to Google Maps; if you're not near enough to your phone to hear the ringer, it can turn the ringer volume up and make your phone ring. One of the features is Activity Log, which enables you to see what the app does, when it does it, and who is using it.

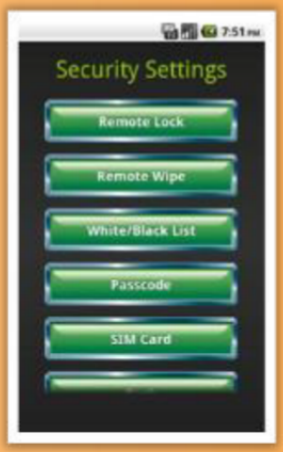


FIGURE 16.34: Wheres My Droid Screenshot

iHound

Source: https://www.ihoundsoftware.com

iHound is an Android device tracking tool that allows you to track your mobile using its GPS and WiFi, 3G, or Edge signals built into your devices to determine its location. Using its tracking website, you can track the location of your device, remotely lock your phone, and remotely erase important personal information such as: SMS messages, contacts, phone call logs, photos, videos, and/or SD storage data. You can also set Geofencing location alerts by its intuitive mobile website optimized for iPhone, iPod Touch, and Android phones. You can track multiple devices on multiple platforms and set up **Geofences**.

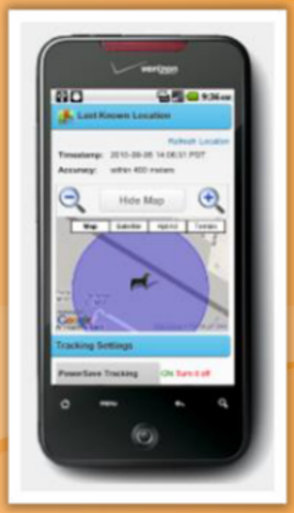


FIGURE 16.35: iHound Screenshot

GadgetTrak Mobile Security

Source: http://www.gadgettrak.com

GadgetTrak Mobile Security tool helps you to **moderate** the risk of mobile device loss or theft. It allows you to track its location, back up data, and even wipes the data in the device remotely. With the combination of GPS, Wi-Fi positioning, and cell tower triangulation, you can easily track the location of your device. If your device is lost or stolen, you can remotely enable a piercing alarm, even if it's in silent mode. Once **tracking** is activated, the software settings cannot be modified unless deactivated.

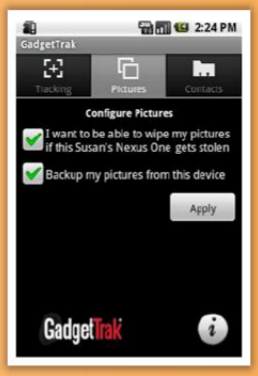


FIGURE 16.36: GadgetTrak Mobile Security

Total Equipment Protection App

Source: https://protection.sprint.com

Total Equipment Protection App is an Android device tracking tool that allows you to **find**, **repair**, and **replace your phone**, whether it is dead or lost. It also comes with online features that protect your existing handset. When you lose the phone, you can map the exact location with directions on how to get there. It sounds the alarm when the phone is misplaced by its alarm even when it is on silent mode. You can choose to remotely lock a misplaced phone or erase your contacts and you can even synchronize and restore the lost phone after its recovery or can get a new phone.

FIGURE 16.37: Total Equipment Protection App Screenshot

AndroidLost.com

Source: http://www.androidlost.com

AndroidLost.com is an online service that allows you to find your lost phone. You don't need to install the **AndroidLost** on the phone but you can push the AndroidLost app to your phone from Google Market and initiate the connection to Google servers by sending an SMS with the message "**Androidlost register**" to your phone when its lost to find its location and tracking. Sound alerts can be enabled even when the phone is in silent mode from your PC. You can control more than one phone from your account.

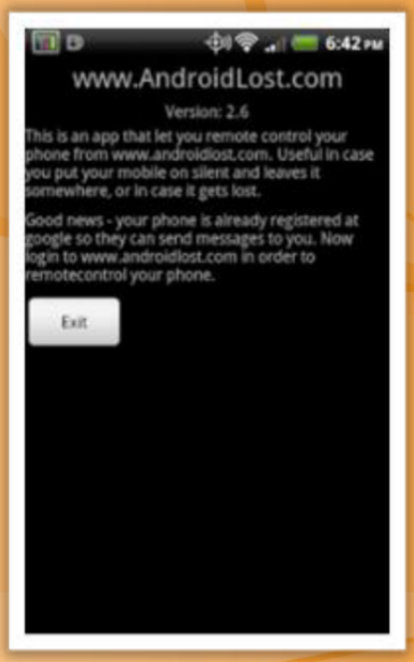
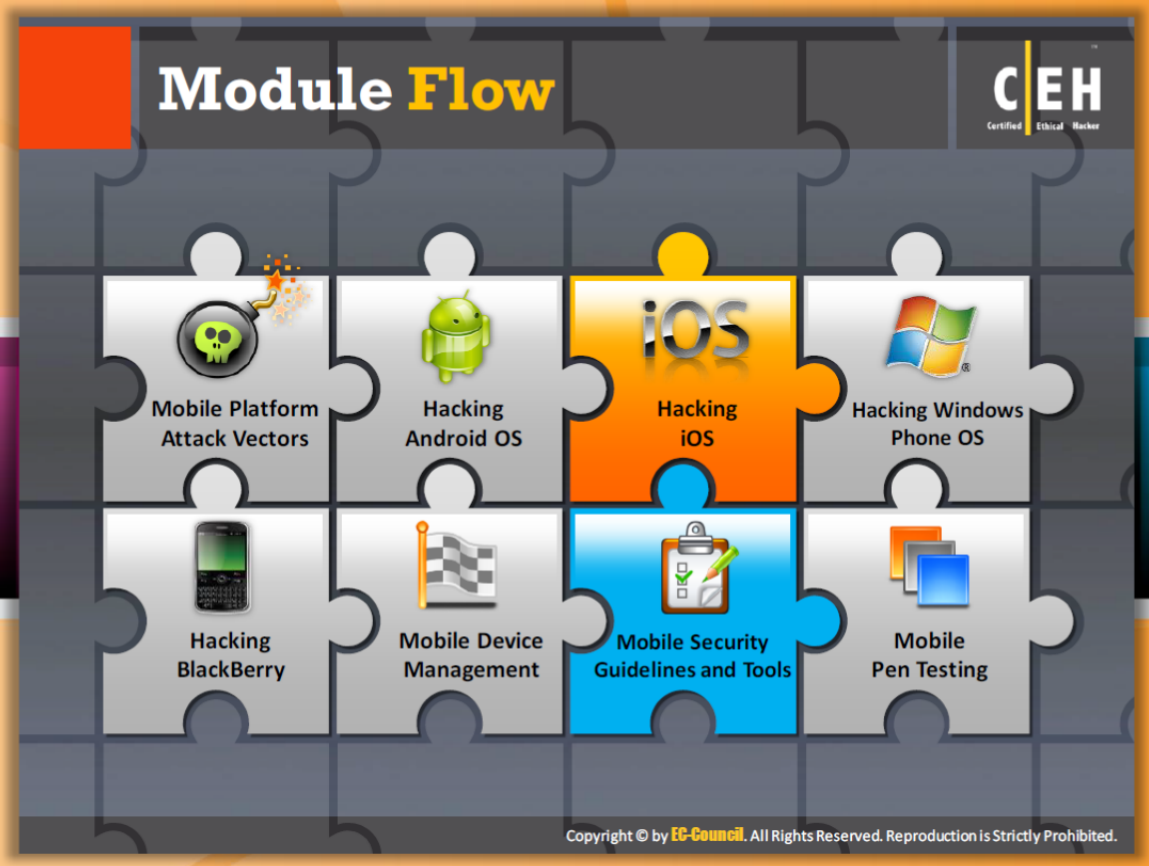


FIGURE 16.38: AndroidLost.com Screenshot



Module Flow

iOS is a mobile operating system developed by Apple. Apple does not license iOS for installation on non-Apple hardware. The increasing use of Apple devices for many purposes has grabbed the attention of attackers. Attackers are concentrating on hacking iOS so that they can gain access to Apple devices at the root level.

Mobile Platform Attack Vectors	Hacking BlackBerry
Hacking Android iOS	Mobile Device Management
Hacking iOS	Mobile Security Guidelines and Tools
Hacking Windows Phone OS	Mobile Pen Testing

This section introduces you to the Apple iOS and focuses on hacking iOS. This section describes iOS attack vectors such as jailbreaking and types of jailbreaking, and also covers the guidelines to be followed in order to secure iOS devices.

Security News

Researchers Hack iPhone Running Latest Apple iOS, Steal Data

Source: http://www.computerworld.in

White-hat hackers broke into the developer version of iOS 6, meaning Apple's new iPhone 5 could be vulnerable.

Researchers have broken into an iPhone 4S running the latest version of Apple iOS, making it possible to exploit the same **vulnerability** in the iPhone 5.

The white-hat hackers Joost Pol and Daan Keuper showed how they were able to steal contacts, browsing history, photos and videos to win $30,000 in the mobile **Pwn2Own** contest Wednesday at EUSecWest in Amsterdam, IT World reports.

Because the hacked iPhone was running a developer version of iOS 6, it's likely the same vulnerability could be used to break into an iPhone 5 or the latest iPad and iPod Touch devices.

The WebKit browser exploit took only a few weeks to make, the researchers told IT World. Using the malicious code in a website would enable a cybercriminal to bypass the security mechanisms in Safari to **gain access** to the phone's data.

WebKit is a layout engine used by browsers to render Web pages. The open source technology is used in the Safari Web browser in iOS and in Google's Chrome, which recently became the default browser for Android.

The Dutch researchers are not the first penetrate the iPhone's defenses through WebKit, said Chenxi Wang, an analyst for Forrester Research. Hackers typically target WebKit because Apple does not use a number of standard security practices in using the engine.

Apple has not said why, but it could be related to phone performance and battery life. In addition, Apple doesn't vet **code executed** on the browser, like it does apps before allowing them to be offered to iPhone users.

"This opens doors to remote exploitation," Wang said. "But to [Apple's] credit, we haven't seen a lot of that going on, which is actually quite impressive."

Wang does not believe the **risk** of the latest vulnerability is very high. That's because a cybercriminal would have to find a way to get iPhone users to a compromised site. A hacker could inject malicious code into a popular Web site, but this would also be difficult.

"It's certainly possible and certainly is a threat, but I don't see it becoming a massively popular way of attacking iPhone users," he said.

The Dutch researchers held back some of the details of their work, in order to prevent giving **cybercriminals** a hacking roadmap to the iPhone.

"Apple will have to come up with an update and then people need to upgrade as fast as possible," Pol told IT World.

Speed in plugging the hole is key to reducing risk, said Peter Bybee, president and chief executive of cloud security provider Security On-Demand.

"Whether you're likely to be attacked depends on how long the gap will be between when Apple fixes the problem and attackers repeat the researcher's success," Bybee said. "Just because the exploit is shared only with the vendor doesn't mean that it won't get out into the open market. There was enough detail in how they found the exploit and used it that it could be replicated by an experienced malware creator."

Other participants in the hacker contest demonstrated breaking into the Samsung Galaxy S3 via its near field communication (NFC) technology. The researchers from security company MWR Labs were able to beam an exploit from one Galaxy S3 to another.

Once the malicious app is installed in the receiving phone, a hacker would have full access to the phone's data, Tyrone Erasmus, a security researcher at MWR told IT World. The app runs in the background, making it invisible to the phone's user.

The exploit targets vulnerability in the document viewer application that comes as a default app in the Galaxy S2, S3 and some HTC phones. The flaw enables a hacker to steal text messages, emails, contact information and other data.

The researchers said the vulnerability, which also exists in the Galaxy S2, could be exploited by malware sent via email, the MWR team said. The researchers also won $30,000 for the hack.

Zero Data Initiative by Hewlett-Packard's DVLabs organized the competition. DVLabs will send details of the hacks to Apple and Samsung, respectively.

By Antone Gonsalves

http://www.computerworld.in/news/researchers-hack-iphone-running-latest-apple-ios-steal-data-29822012

Apple iOS

iOS is the Apple mobile's operating system established for its **iPhones**. It maintains and sustains other Apple devices such as iPod Touch, iPad, and Apple TV. Using the Mac OS X, the iOS operating system is fabricated. The user interface is based on the concept of direct manipulation, using multi-touch **gestures**. This has many other options and features using which daily work becomes easy and this can be updated on your iPhone, iPad, or iPod Touch using **Wi-Fi** and other wireless networks.

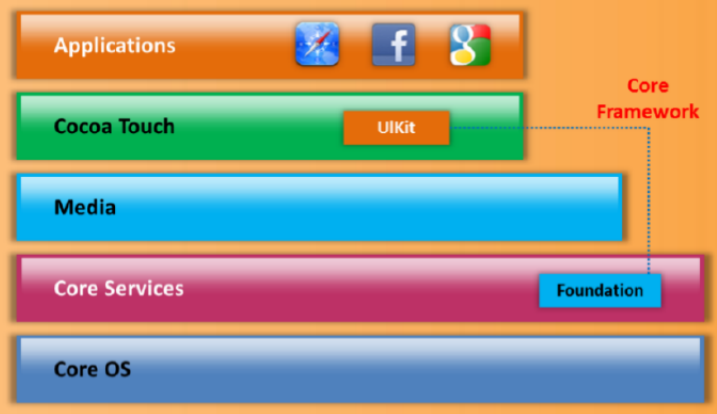


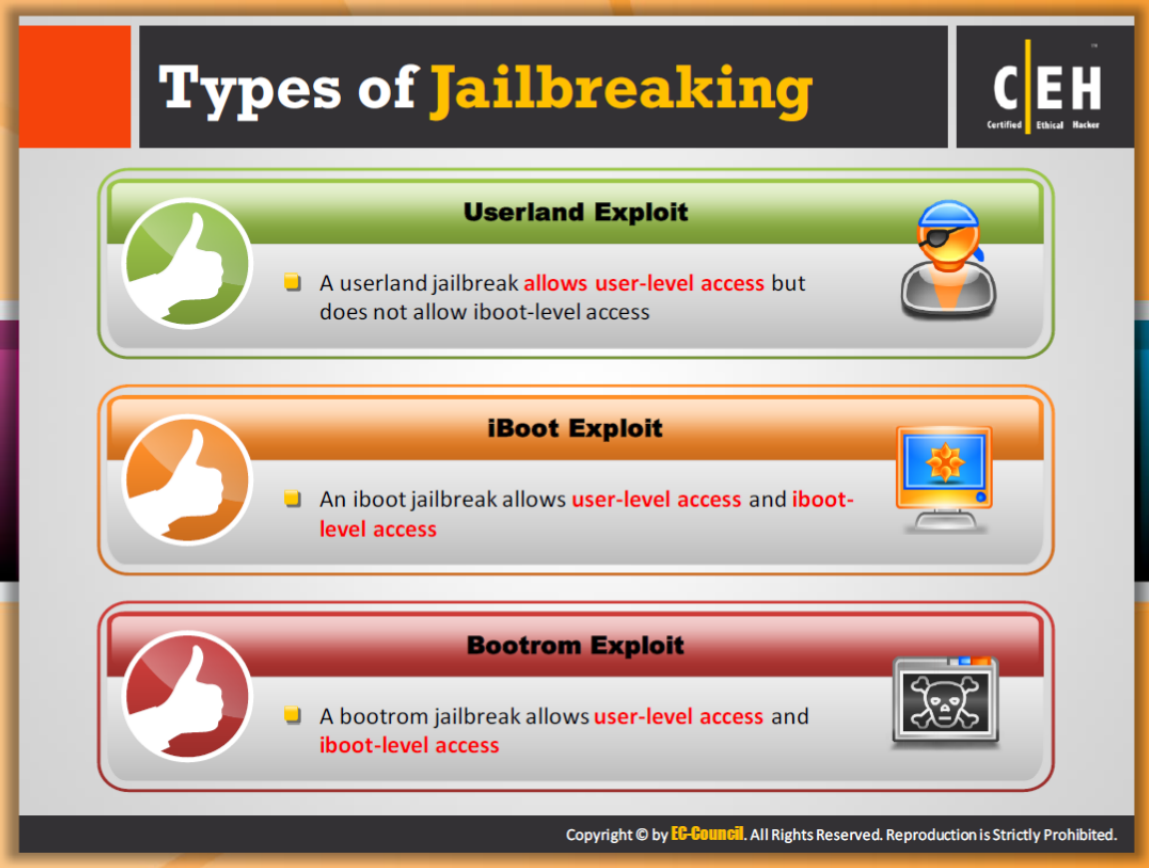
FIGURE 16.39: Apple Ios Screenshot

Jailbreaking iOS

Jailbreaking is a method of **getting control** of the **iOS operating system** that is used on Apple devices. It relaxes the device from the barriers of dependencies on exclusive Apple source applications and allows the user to use third-party apps unavailable at the official app store. It is accomplished by installing a modified set of kernel patches that allow you to run third-party applications not signed by the **OS vendor**. It is used to add more functionality to standard Apple **gadgets**. It can also provide root access to the operating system and permits download of third-party applications, themes, extensions, etc. This removes sandbox restrictions, which enables malicious apps to access restricted mobile resources and information.

Jailbreaking, like rooting, also comes along with many security and other risks to your device including:

- Voids your phone's warranty
- Poor performance
- Bricking the device
- Malware infection



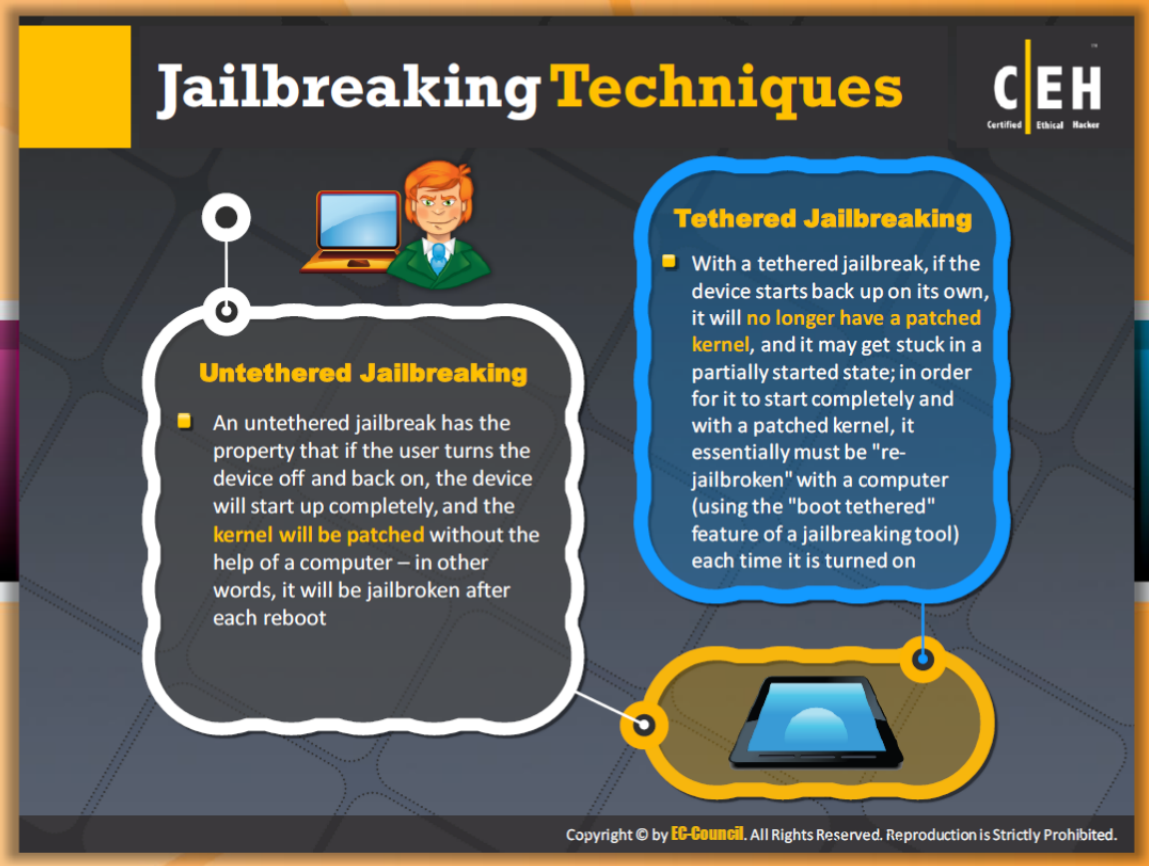
Types of Jailbreaking

When the device starts booting, it loads **Apple's** own **iOS** at start, but to get more apps from third parties, the device must then be broken and have the kernel patched each time it is turned on. There are three types of jailbreaking methods used:

Userland Exploit: A userland jailbreak allows user-level access but doesn't allow **iboot-level** access. This type of exploit cannot be tethered as it cannot have recovery mode loops. These can be patched by Apple. The userland exploits use a loophole in the system application to gain control of that application. This exploit can only give control to the **filesystem**. This type of exploit can access non-vital code in the application and is user friendly and platform independent.

iBoot Exploits: An iBoot jailbreak allows file system and iboot level access. This type of exploit can be semi-tethered if the device has a new **bootrom**. This is mostly used to reduce low-level iOS controls. This exploit method takes the help of the hole in iBoot to delink the code signing appliance and then the customer can download required applications. Using this method users configure the mobile to accept **custom firmware** and probably jailbreak more.

Bootrom Exploits: A bootrom jailbreak can break all the **low-level authentications** such as providing filesystem, iBoot, and NOR access (custom boot logos). This process finds a hole in the application to discard the signature checks. It can't be corrected by Apple. A bootrom jailbreak allows user-level access and iBoot-level access. These cannot be patched by Apple.



Jailbreaking Techniques

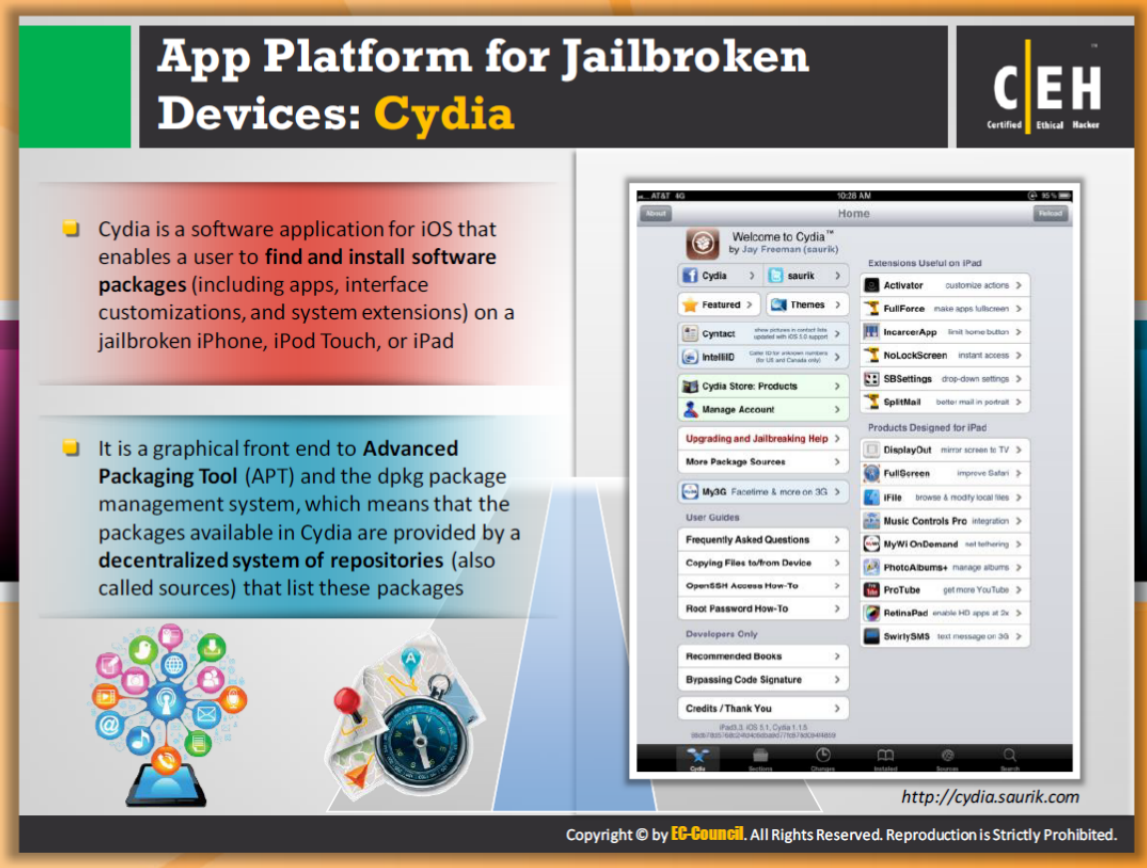
There are two jailbreaking techniques:

Untethered Jailbreaking

Untethered jailbreak is a method of **rebooting** the **mobile device** without connecting it to the system every time you boot. If the battery of the device is spoiled, after changing it boots as usual. Some jailbreak solutions are greenpois0n, PwnageTool, limera1n, and sn0wbreeze.

Tethered Jailbreaking

With a tethered jailbreak, if the device starts back up on its own, it will no longer have a patched kernel, and it may get stuck in a partially started state; in order for it to start 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.

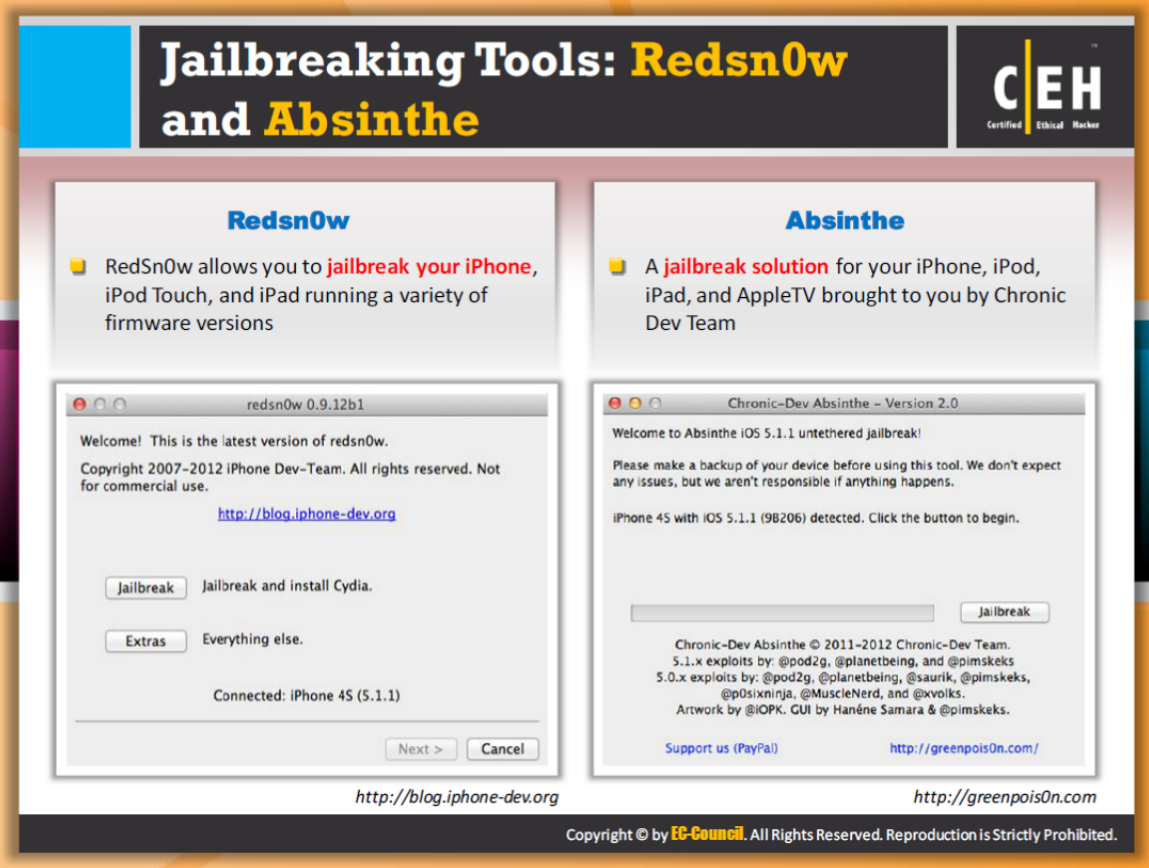


App Platform for Jailbroken Devices: Cydia

Source: http://cydia.saurik.com

Cydia is a software application specifically designed for iOS enabled services for devices to jailbreak that facilitates a user to install software on iPhone, iPod Touch, iPad, etc. It has many different applications, extensions, themes, features, and **customizations**. It is a graphical front end to **Advanced Packaging Tool (APT)** and the dpkg package management system, which means that the packages available in Cydia are provided by a decentralized system of repositories (also called sources) that list these packages.

FIGURE 16.40: Cydia Screenshot



Jailbreaking Tools: Redsn0w and Absinthe

Redsn0w

Source: http://blog.iphone-dev.org

RedSn0w allows you to jailbreak your **iPhone**, **iPod Touch**, and **iPad** running a variety of firmware versions. This is developed by the iPhone Dev Team. It supports Windows and Mac OS X operating systems to jailbreak iOS devices, both tethered and untethered.

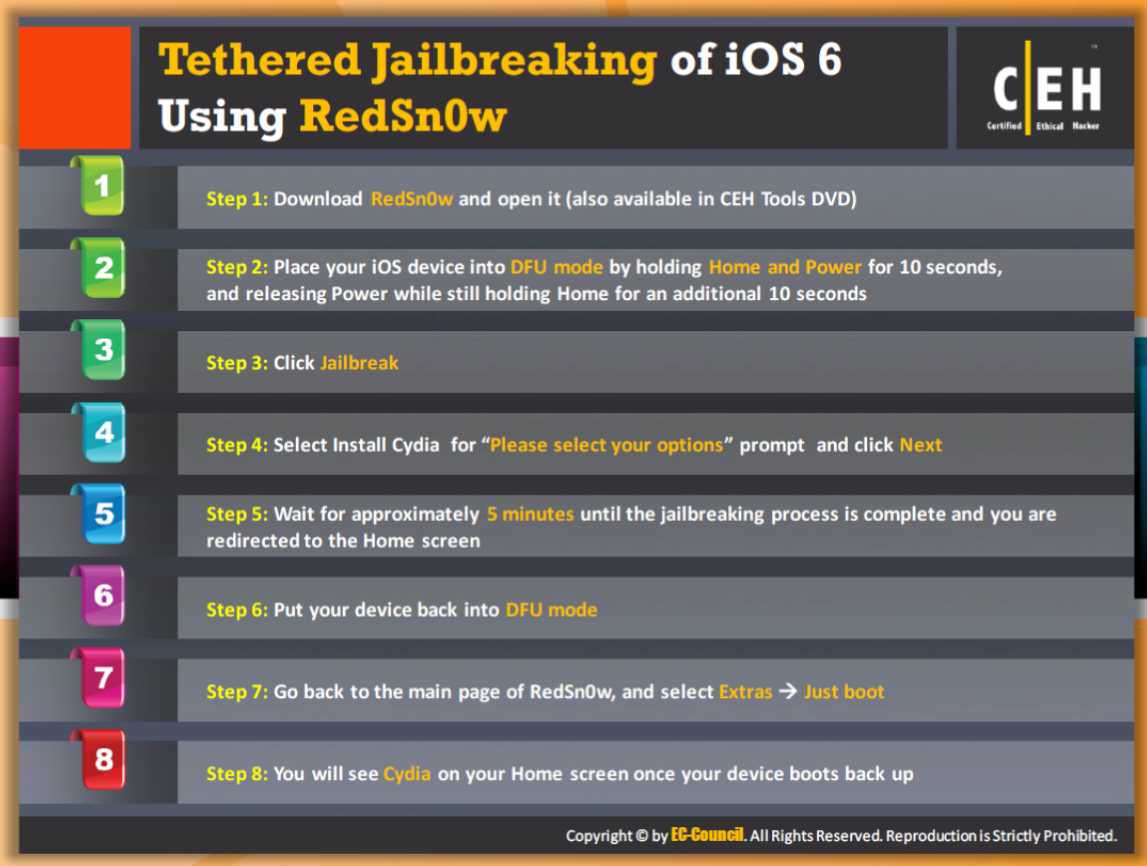
FIGURE 16.41: Redsn0w Screenshot

Absinthe

Source: http://greenpois0n.com

Absinthe is a jailbreak solution for your **Apple mobile devices**, including the iPhone, iPad, iPod Touch, and AppleTV brought to you by Chronic Dev Team; their aim is to develop iOS untethered jailbreak toolkits.

FIGURE 16.42: Absinthe Screenshot



Tethered Jailbreaking of iOS 6 Using RedSn0w

As mentioned previously, **Redsn0w** can be used for both tethered and untethered jailbreaking. Let's discuss the process or steps involved in tethered jailbreaking of iOS 6 using RedSn0w:

Step 1: Download RedSn0w and open it (also available in CEH Tools DVD).

Step 2: Place your iOS device into DFU mode by holding Home and Power for 10 seconds, and releasing Power while still **holding Home** for an additional 10 seconds.

Step 3: Click **Jailbreak**.

Step 4: Select **Install Cydia** under the **Please select your options** prompt and click **Next**.

Step 5: Wait for approximately 5 minutes until the **jailbreaking** process is complete and you are redirected to the Home screen.

Step 6: Put your device back into DFU mode.

Step 7: Go back to the main page of RedSn0w, and select **Extras → Just boot**.

Step 8: You will see **Cydia** on your Home screen once your device boots back up.

Jailbreaking Tools: Sn0wbreeze and PwnageTool

Sn0wbreeze

Sn0wBreeze is a jailbreaking tool for **Windows OS** to create a custom **Pre-Jailbroken** iOS firmware file that must be restored to your iPhone, iPod Touch, or iPad for it to become jailbroken. It allows **iPhone** unlockers to update to the latest firmware without updating their **baseband** in the process. This gives you full control over your jailbreak, allowing you to customize advanced options such as your root partition size.

FIGURE 16.43: Sn0wbreeze Screenshot

PwnageTool

Pwnage is **a jailbreaking** tool that allows you to unlock and create a custom IPSW, thus allowing you to update your firmware while still preserving the baseband for unlocking. Even if your baseband isn't **unlockable**, you may want to preserve your **baseband** in case a future unlock is found. This tool is compatible with Mac OS.

FIGURE 16.44: PwnageTool Screenshot

Jailbreaking Tools: LimeRa1n and Jailbreakme

LimeRa1n

Source: http://www.limera1n.com

LimeRa1n is a **jailbreaking** tool invented by a **GeoHot** (professional hacker) to halt Chronic Dev from releasing a bootrom exploit called SHAtter. One of the features of this tool enables you to switch between jailbreaking methods and it supports the Windows and **Mac OS X** operating systems.

FIGURE 16.45: LimeRa1n Screenshot

Jailbreakme

Source: http://www.jailbreakme.com

JailbreakMe is a tool that allows you to jailbreak your iPhone, iPod Touch, or iPad through **online services**. It is used to provide a jailbreak for the iPad 2 **untethered**.

FIGURE 16.46: Jailbreakme Screenshot

Jailbreaking Tools: Blackra1n and Spirit

Blackra1n

Source: http://blackra1n.com

Blackra1n is a jailbreaking tool that allows you to jailbreak devices such as an iPhone, iPod, or iPad on **firmwares**. This can work on all devices without having to make **adjustments** in advance in the software. It works on both Windows and **Mac OS**. It is designed by Geohot.

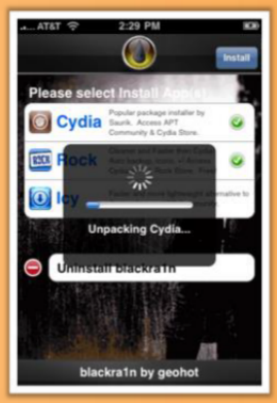


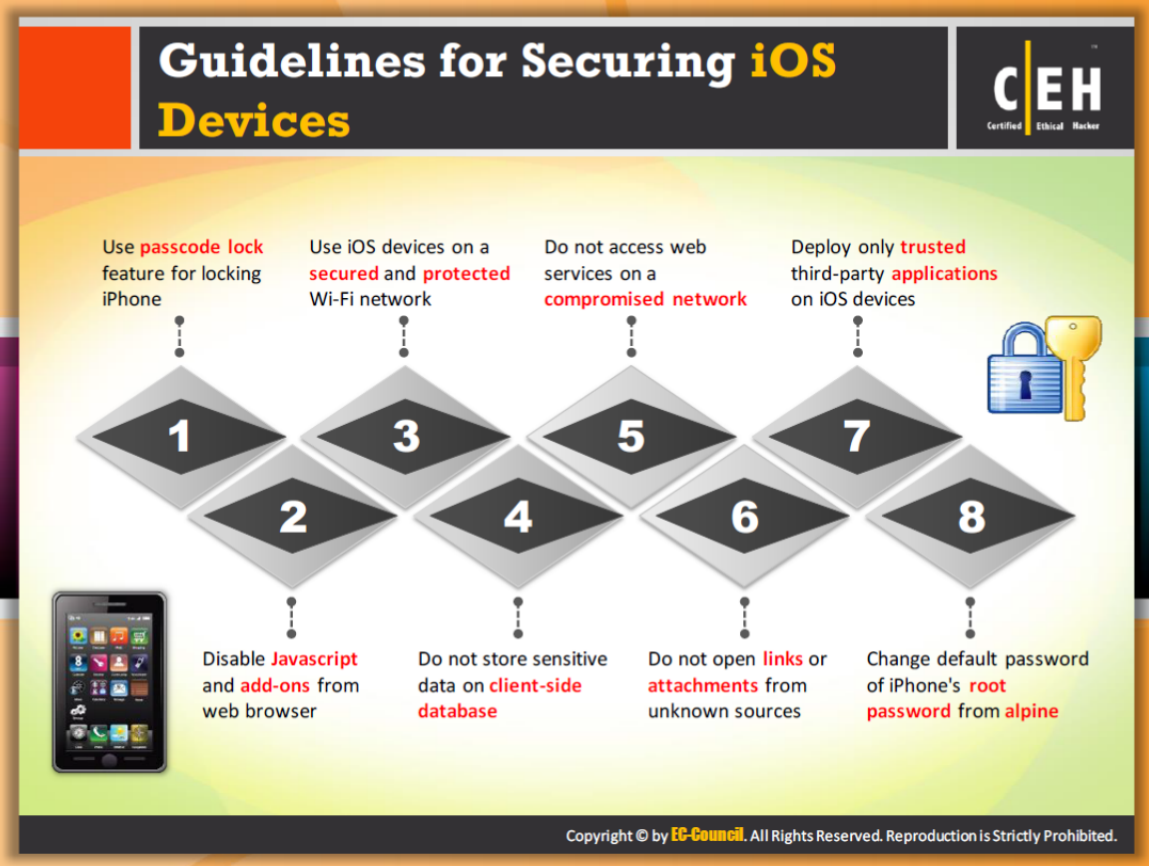
FIGURE 16.47: Blackra1n Screenshot

Spirit

Source: http://spiritjb.com

Spirit is a jailbreaking tool that allows you to jailbreak devices that are **untethered**. It can jailbreak the iPad, iPhone, and iPod touch on certain **firmware versions**. It is not a carrier unlock.

FIGURE 16.48: Spirit Screenshot



Guidelines for Securing iOS Devices

Guidelines for security iOS determine the **course of action** that helps in enhancing the security of iOS devices. These guidelines are not **mandatory** to apply, but help in protecting iOS devices from being attacked. The following are a few guidelines for security iOS:

- Use passcode lock feature for **locking iPhone**
- Disable JavaScript and add-ons from web browsers
- Use iOS devices on a secured and protected **Wi-Fi** network
- Do not store sensitive data on a client-side database
- Do not access web services on a **compromised** network
- Do not open links or **attachments** from unknown sources
- Deploy only trusted **third-party** applications on iOS devices
- Change default password of iPhone's root password from Alpine

Guidelines for Securing iOS Devices (Cont'd)

Guidelines that are to be followed by every user in order to secure iOS devices against attacks include:

- Do not jailbreak or root your device if used within **enterprise environments**
- Configure Find My iPhone and utilize it to wipe a **lost** or **stolen device**
- Enable Jailbreak detection and also protect access to iTunes AppleID and Google accounts, which are tied to **sensitive data**
- Disable **iCloud services** so that sensitive enterprise data is not backed up to the cloud (note that cloud services can back up documents, account information, settings and messages)
- Along with this follow the common **security guidelines** for all the mobile devices outlined in the later slides

iOS Device Tracking Tools

Find My iPhone

Source: https://itunes.apple.com

Find My iPhone **iOS Device** Tracking Tool allows you to track a lost or **misplaced** mobile, iPhone, iPad, iPod touch, or Mac. This allows you to use another iOS device to find it and **protect** your data. To use this, you need to install the app on another iOS device, open it, and sign in with your **Apple ID**. It helps you locate your **missing device** on a map, play a sound, and even display a message, remotely.

FIGURE 16.49: Find My iPhone Screenshot

iHound

Source: https://www.ihoundsoftware.com

iHound is a iOS device tracking tool that allows you to track your device by simply turning on iHound; minimize it and let it run. You can even delete it from the fast app switching bar. It can still locate your phone anytime, anywhere.

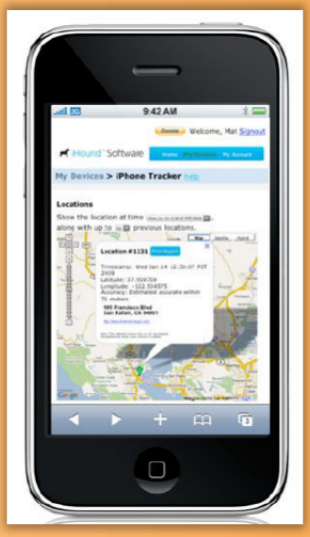


FIGURE 16.50: Find My iPhone Screenshot

GadgetTrak iOS Security

Source: http://www.gadgettrak.com

GadgetTrak iOS Security is an **iOS device tracking tool** that allows you to recover your iPhone, iPad, or iPod touch by using the ability to track your device by using GPS, Wi-Fi positioning, and

cell tower triangulation to **pinpoint location**. Using the built-in cameras, you can collect crucial evidence to help catch the thief. When tracking occurs, you'll receive an email with detailed information about its current location. Once tracking is **activated** the software settings cannot be modified unless deactivated. When tracking data is being **transmitted** from your device, a secure SSL connection is used. Only you can access your location reports and camera. All images, network information, and location data are sent directly to you from your device.

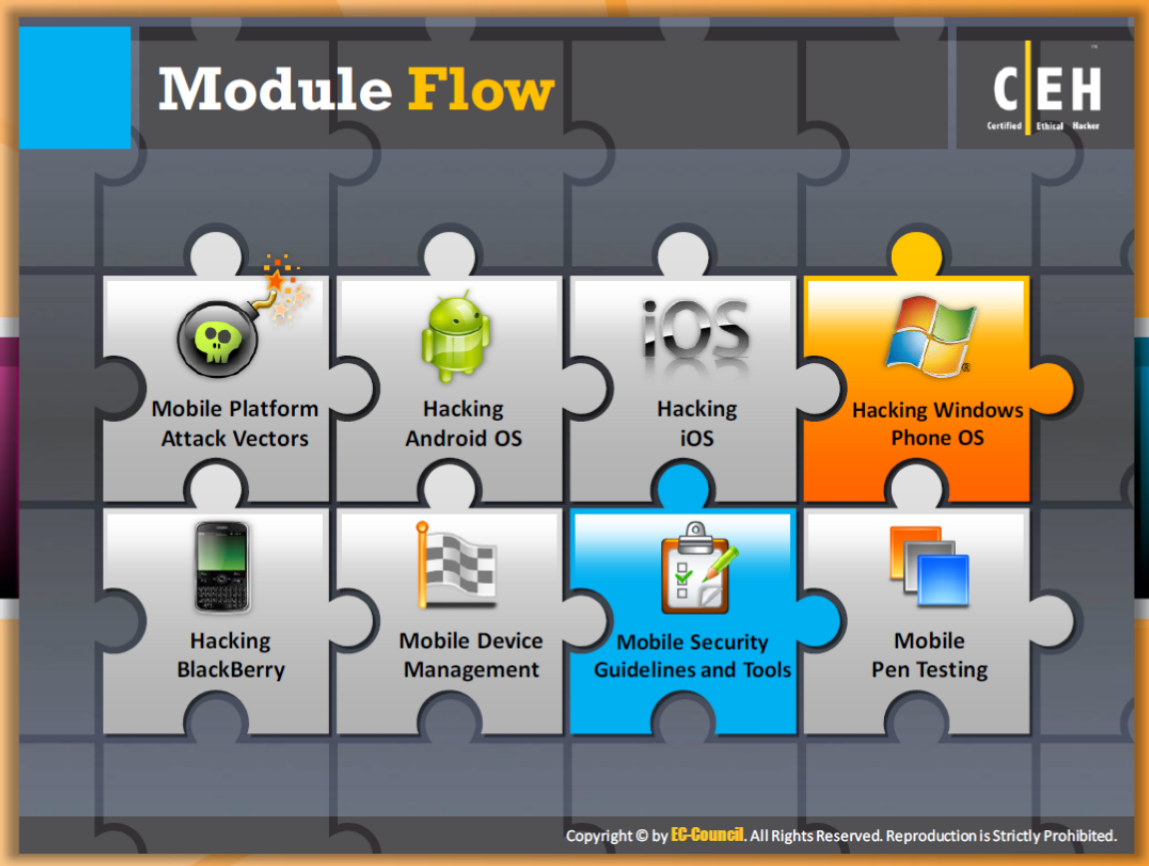
FIGURE 16.51: GadgetTrak iOS Security Screenshot

iLocalis

Source: http://ilocalis.com

iLocalis iOS device tracking tool allows you to **control your iPhone** from your computer connected to the Internet. If your iPhone has been stolen you can find it with the track feature or even make a remote call or SMS to see the new number if the SIM has been changed. It has many features such as location **tracking** and sharing location with others, remote iPhone control, and SMS commands with backup and remote wipe of data. It has alert zone, push support, and remote audio recording with iPhone lock.

FIGURE 16.52: iLocalis Screenshot

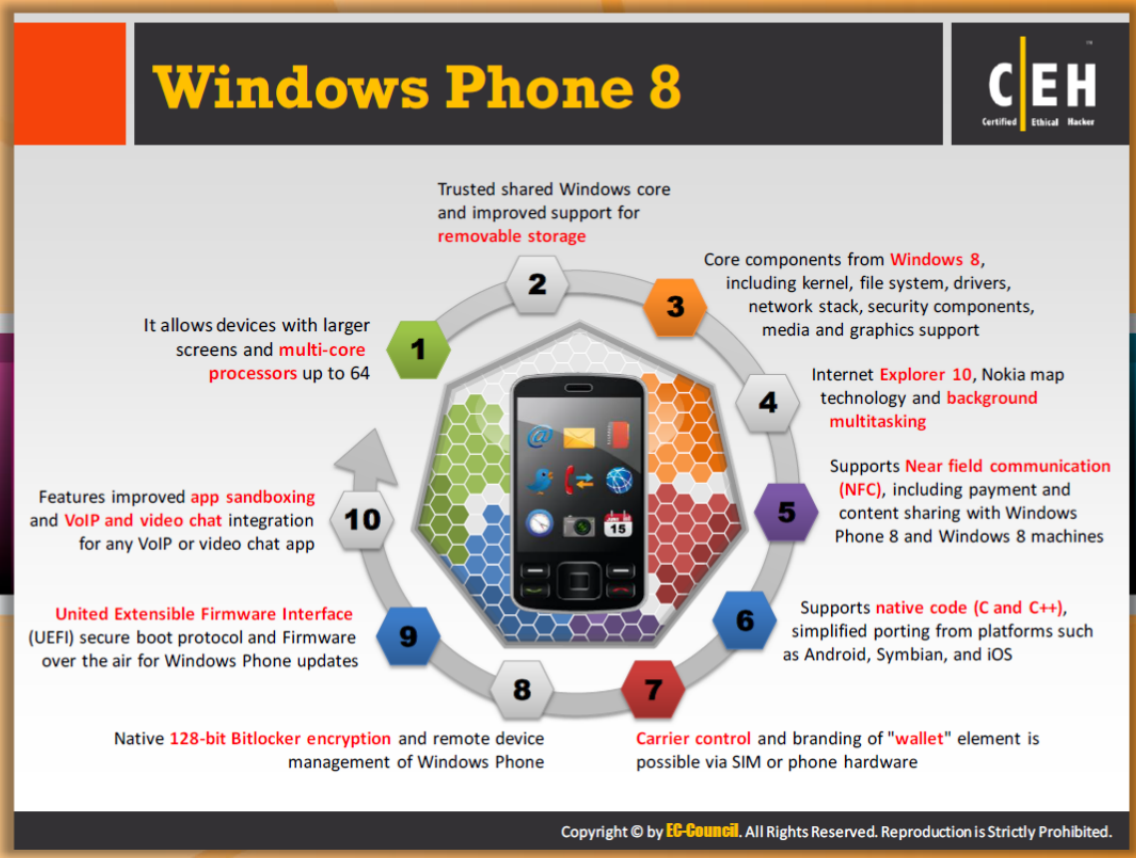


Module Flow

So far, we have discussed how to hack iOS. Now we will discuss hacking the Windows Phone OS. Similar to Apple's iOS, Windows Phone OS is another operating system intended for mobile devices.

Mobile Platform Attack Vectors	Hacking BlackBerry
Hacking Android iOS	Mobile Device Management
Hacking iOS	Mobile Security Guidelines and Tools
Hacking Windows Phone OS	Mobile Pen Testing

This section introduces you to Windows Phone 8 and its architecture and secure boot process.

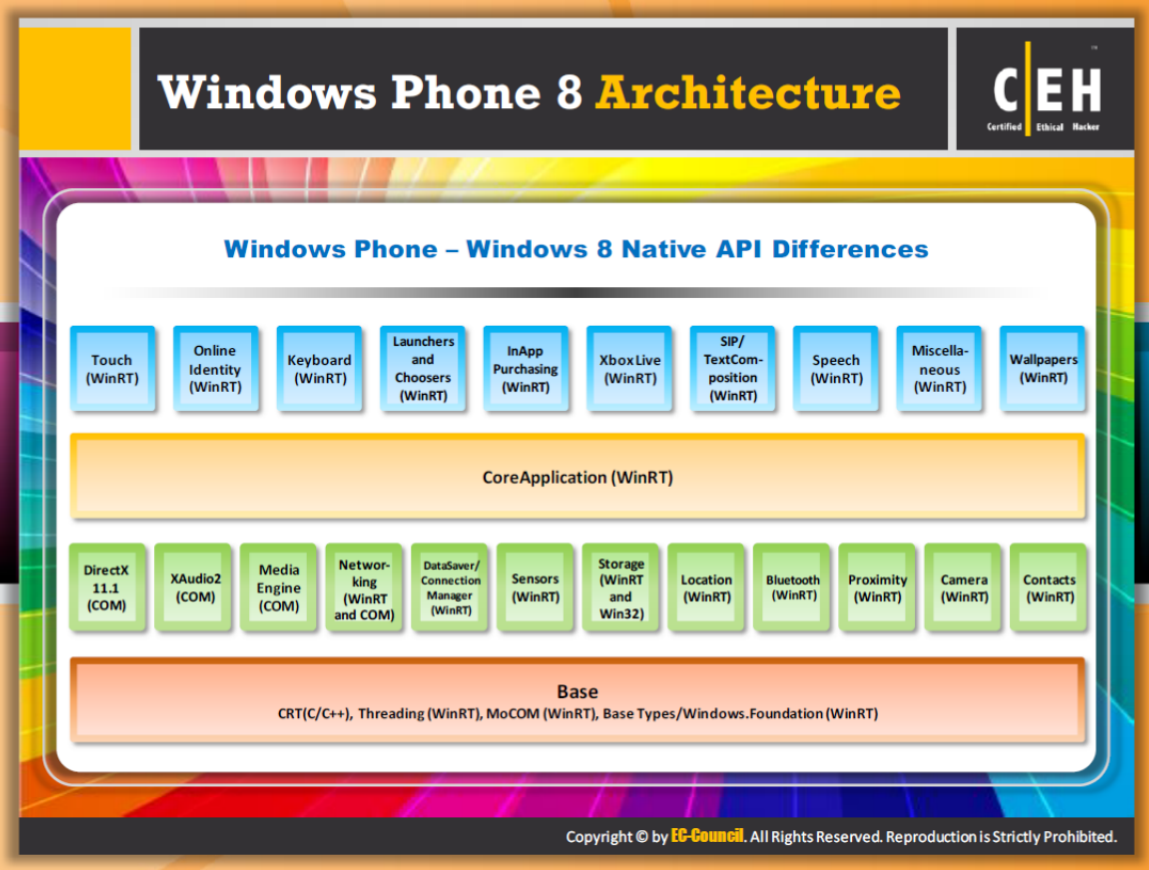


Windows Phone 8

Windows Phone 8 is the second generation operating system developed by Microsoft for Windows Phone. A few **important** points about Windows Phone 8 are as follows:

- It allows devices with larger screens and multi-core processors up to **64 cores**.
- Trusted shared Windows core and **improved** support for removable storage.
- Core components from Windows 8, including kernel, file system, drivers, network stack, security components, media and **graphics support**.
- Internet Explorer 10, Nokia map technology, and background multitasking.
- Supports **Near field communication** (NFC), including payment and content sharing with Windows Phone 8 and Windows 8 machines.
- Supports native code (C and C++), simplified porting from **platforms** such as Android, Symbian, and iOS.
- Carrier control and branding of "**wallet**" element is possible via SIM or phone hardware.
- Native 128-bit Bitlocker encryption and remote device management of Windows Phone.
- United Extensible Firmware Interface (UEFI) **secure boot protocol** and Firmware over the air for Windows Phone updates.

- Features improved app sandboxing and VoIP and video chat integration for any VoIP or video chat app.



Windows Phone 8 Architecture

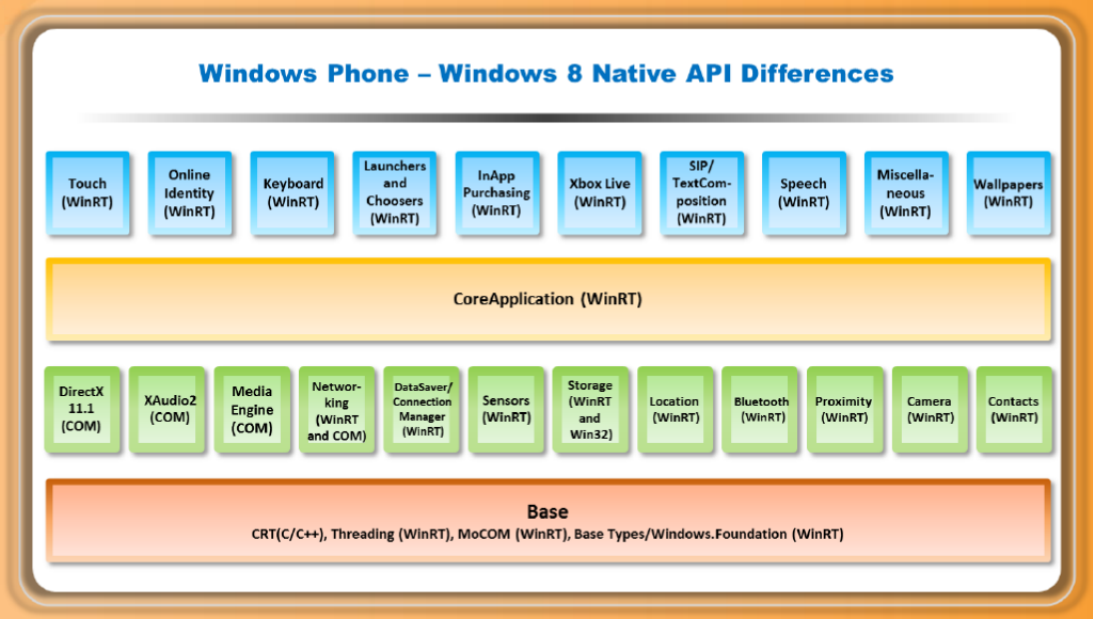
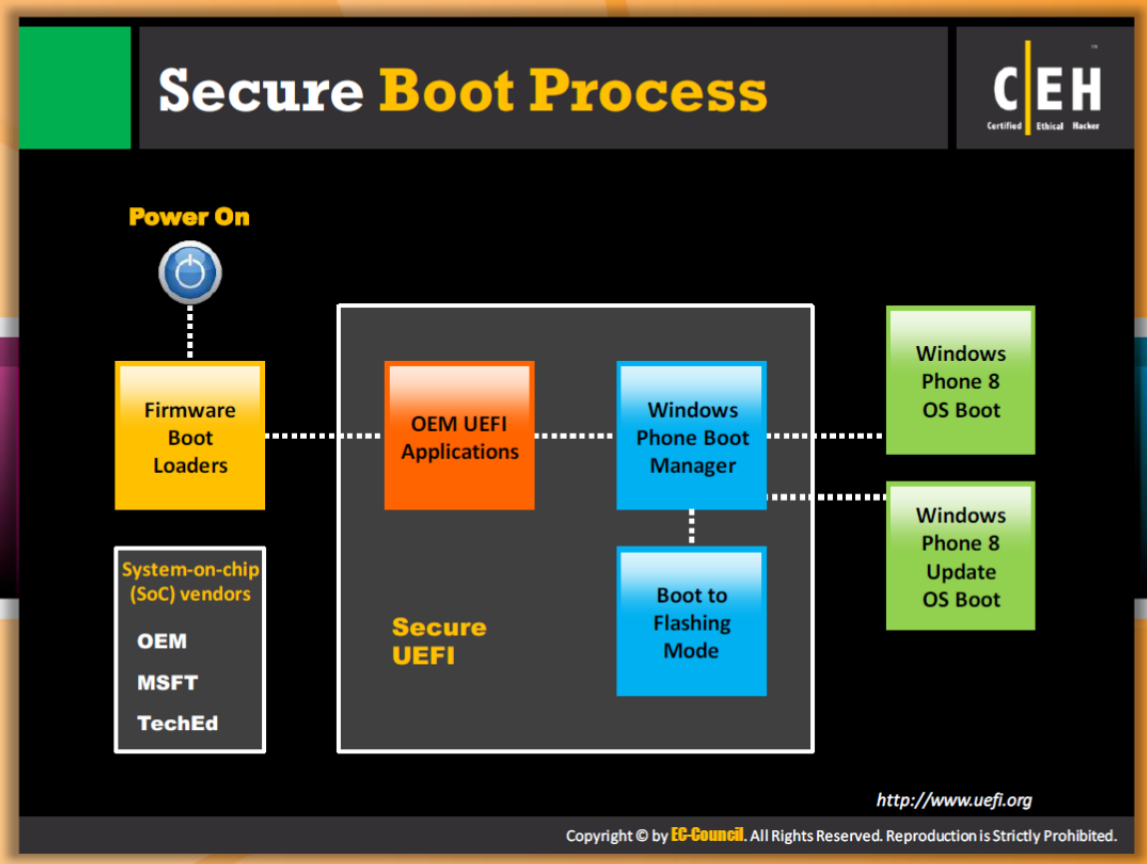


FIGURE 16.53: Windows Phone 8 Architecture



Secure Boot Process

Source: http://www.uefi.org

The goal of the SafeBoot feature of **Windows Phone 8** is to design a SafeBoot process to achieve safe launching of the OS to guarantee only trusted components get loaded. The background of the information system incorporated here is each device gets a distinct key embedded into a chip, along with common keys from Microsoft and the OEM and then the fuse is soldered on the chip.

When you first switch on the power the firmware starts a **Unified Extensible Firmware Interface (UEFI)** background that validates the **hash** of these keys compared to the signatures on the initial boot loaders to confirm the operating environment. In this stage the signatures are compared on the Windows Phone **boot manager** to permit the genuine and trusted applications to start.

Microsoft needs their own binaries along with OEM binaries and they should also have a digital signature signed by Microsoft, which is used to **shield** the application and the boot system from malware. No one can access all the keys that are required to start the system run, and it is not possible to build convenient ROMs and the **signatures** as they may differ from the original signatures.

Microsoft has reduced the OS footprints. All the applications should be run on the same sandbox as third-party marketplace apps, which in turn extend the **customization** of OEM drivers. If any attacker tries to **mitigate** the application with malware it can only access the content inside that sandbox, preventing malware from **gaining access** to the lower system level of the device.

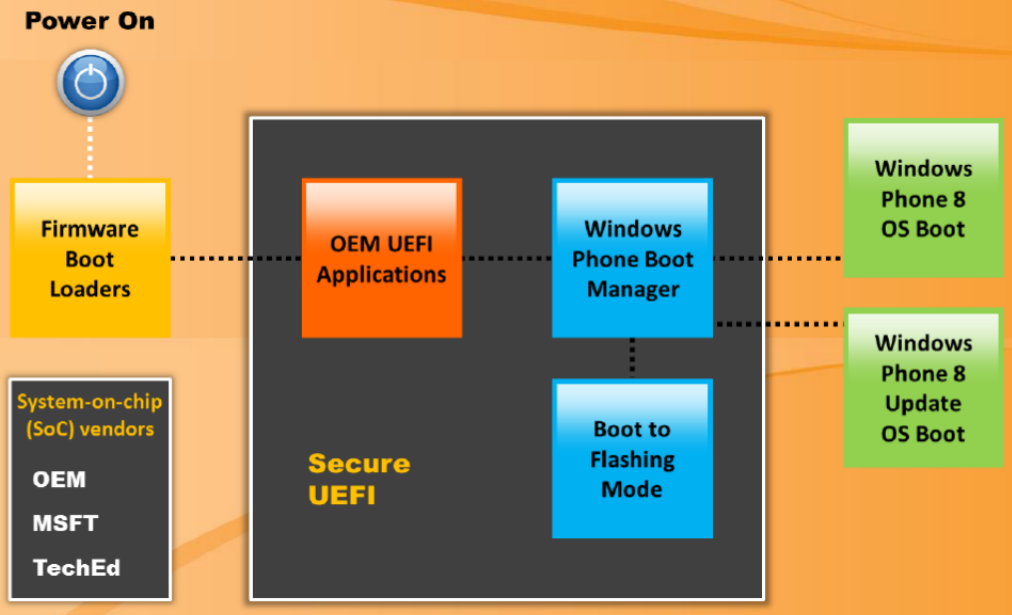
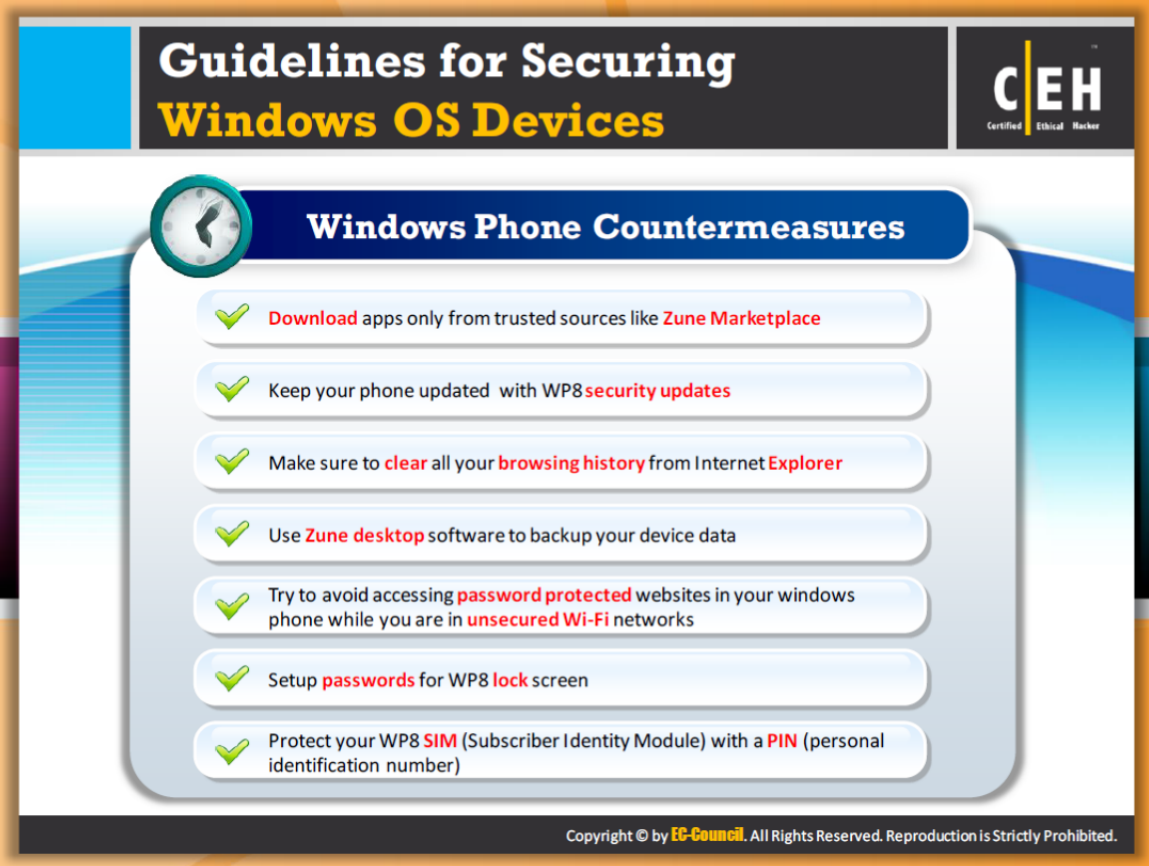


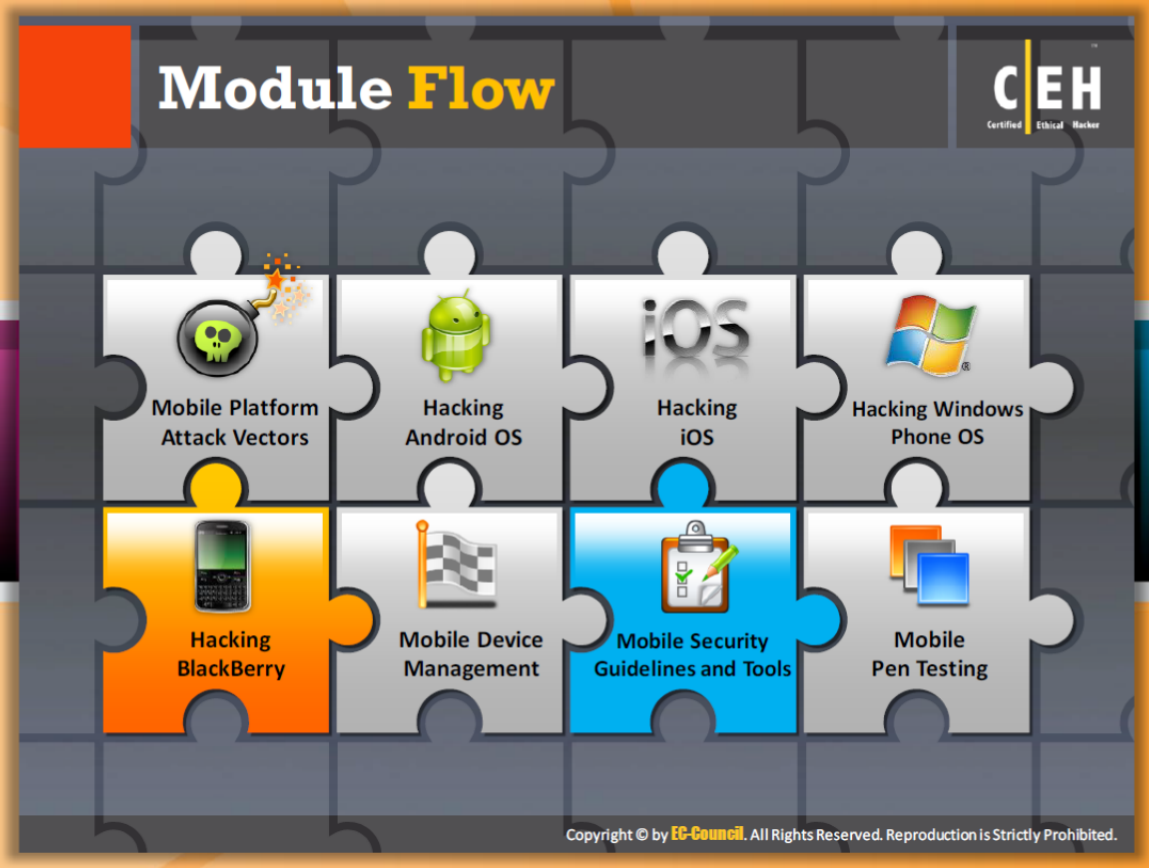
FIGURE 16.54: Secure Boot Process



Guidelines for Securing Windows OS Devices

Guidelines for securing Windows OS devices are the actions to be taken or settings to be changed that are not mandatory but enhance security if applied. The following are a few guidelines that help in **securing** Windows OS devices:

- Download apps only from trusted sources like Zune Marketplace
- Keep your phone updated with **WP8 security** updates
- Make sure to clear all your browsing history from Internet Explorer
- Use Zune desktop software to **backup** your device data
- Try to avoid accessing password protected websites in your windows phone while you are in unsecured Wi-Fi networks
- Setup passwords for **WP8 lock screen**
- Protect your WP8 SIM (Subscriber Identity Module) with a PIN (personal identification number)

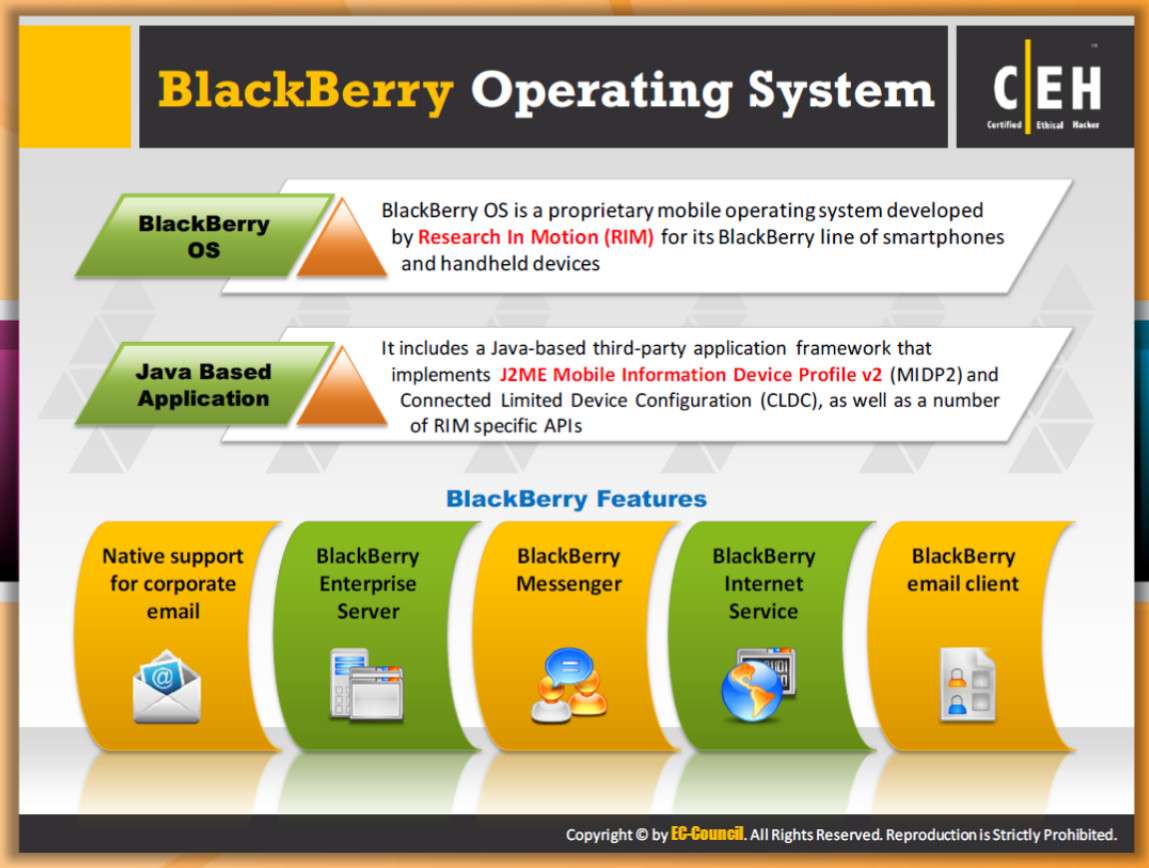


Module Flow

BlackBerry is a brand of wireless handheld devices and service developed by **Research In Motion** (RIM). Attackers are also concentrating on BlackBerry devices.

Mobile Platform Attack Vectors	Hacking BlackBerry
Hacking Android iOS	Mobile Device Management
Hacking iOS	Mobile Security Guidelines and Tools
Hacking Windows Phone OS	Mobile Pen Testing

This section introduces you to the BlackBerry operating system, BlackBerry enterprise solution architecture, and attack vectors. It also covers **guidelines** for securing BlackBerry devices.

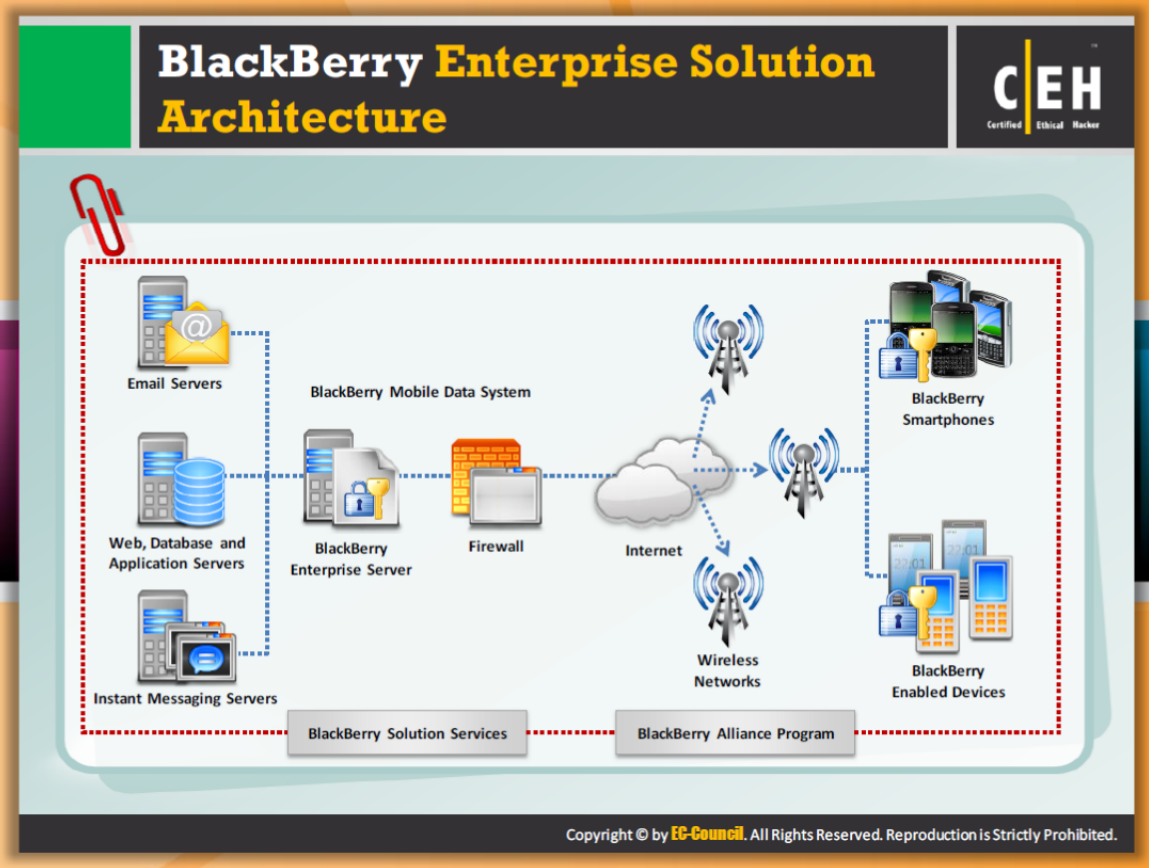


BlackBerry Operating System

BlackBerry OS is a proprietary mobile operating system developed by **Research In Motion (RIM)** for its BlackBerry line of smartphones and handheld devices. It includes a Java-based third-party application framework that implements J2ME Mobile Information Device Profile v2 (MIDP2) and **Connected Limited Device Configuration** (CLDC), as well as a number of RIM specific APIs.

Some of the features of BlackBerry include:

- Native support for corporate email
- BlackBerry Enterprise Server
- BlackBerry Messenger
- BlackBerry Internet Service
- BlackBerry email client



BlackBerry Enterprise Solution Architecture

Blackberry Enterprise Solution allows mobile users to wirelessly access their organization emails and other **business-critical applications** safely and securely. BlackBerry Enterprise Solution Architecture is comprised of six vital elements. They are BlackBerry® Enterprise Server, BlackBerry® Mobile Data System, BlackBerry Smartphones, Devices with BlackBerry® Connect™ software, BlackBerry® Alliance Program, and BlackBerry Solution Services.

The enterprise server, together with **enterprise** messaging and collaboration systems, provides email access to mobile users, enterprise instant messaging, and personal information management tools. Poorly **configured** firewalls increase the risk of attacks. The Web, Database, and Application Server contain **vulnerabilities**. If the attacker detects those vulnerabilities, then he or she can easily carry out an attack and take control over the entire server.

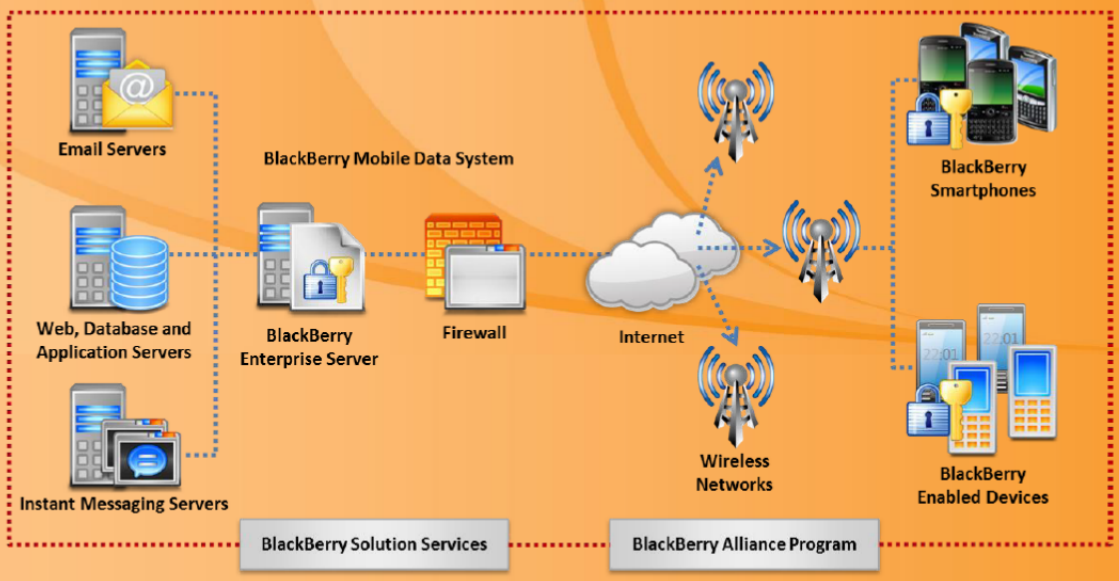
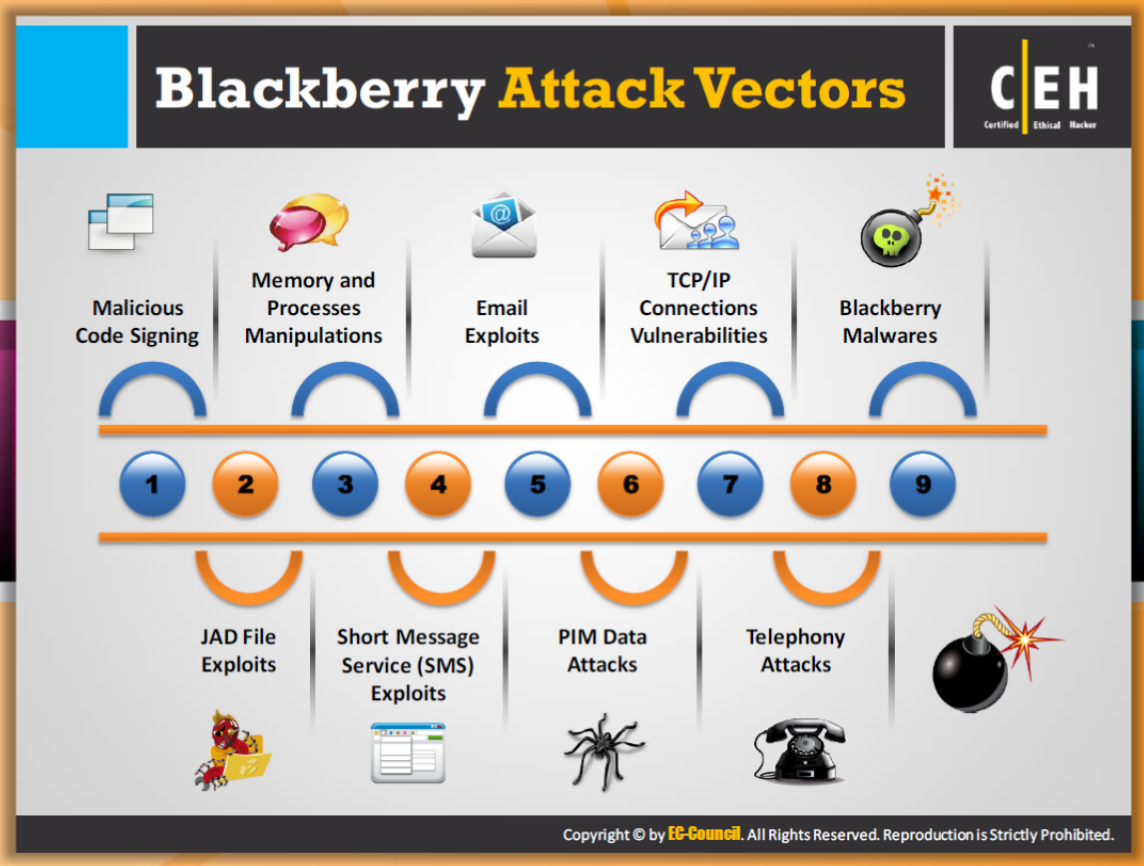


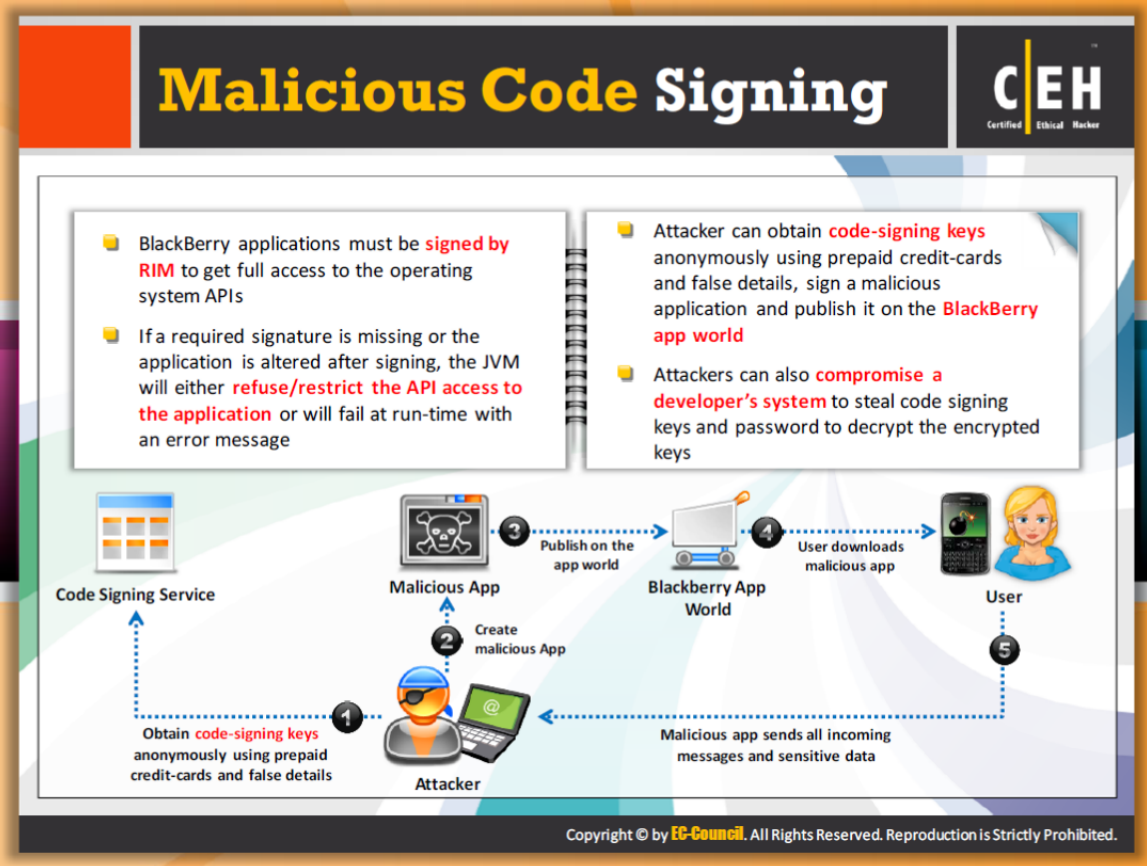
FIGURE 16.55: BlackBerry Enterprise Solution Architecture



BlackBerry Attack Vectors

BlackBerry is prone to many attacks since there are many new tools and methods available for finding potential **vulnerabilities** present on BlackBerry devices. Attack vectors such as luring and attracting users to download **malicious software** on their mobiles, finding website vulnerabilities using tools, etc. are the few techniques used by an attacker for carrying out attacks on **BlackBerry** devices. Apart from these techniques there are many more attack vectors that allow attackers to launch attacks on **BlackBerrys** that include:

- Malicious Code Signing
- Memory and Processes Manipulations
- Email Exploits
- TCP/IP Connections Vulnerabilities
- Blackberry Malwares
- JAD File Exploits
- Short Message Service (SMS) Exploits
- PIM Data Attacks
- Telephony Attacks

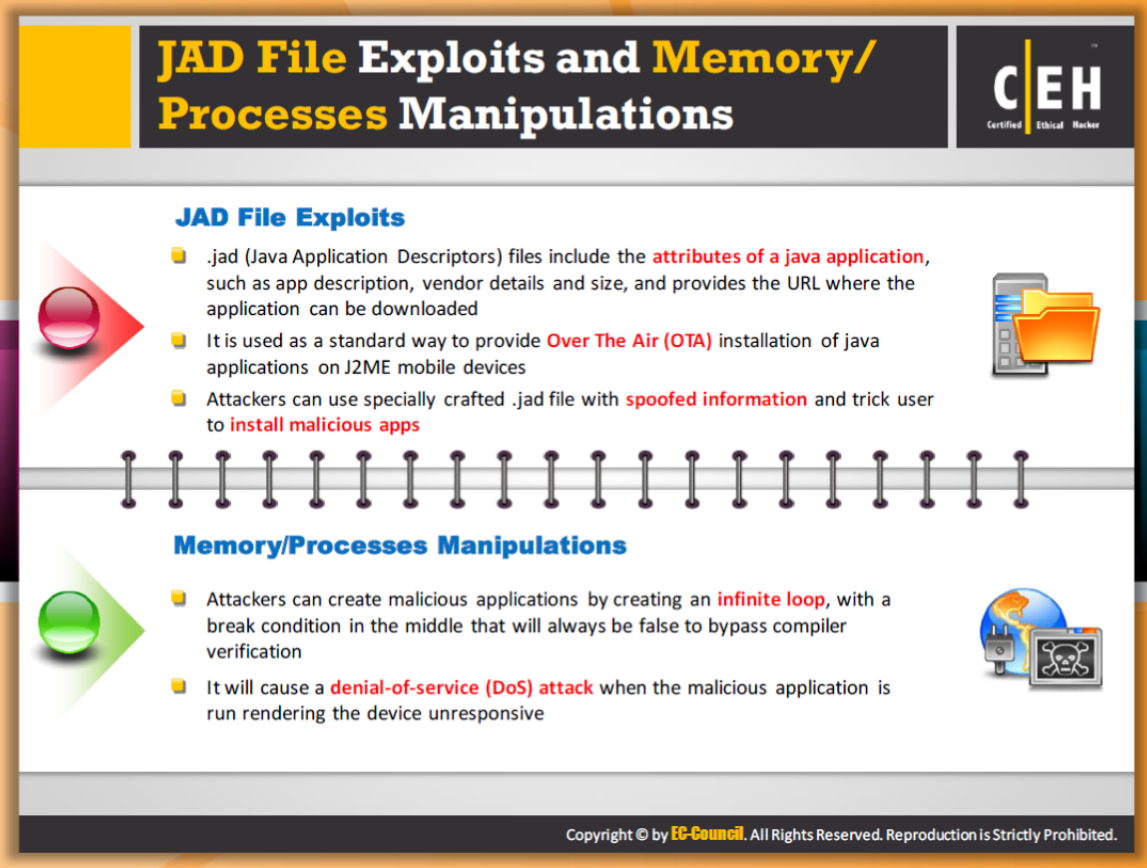


Malicious Code Signing

BlackBerry applications must be signed by RIM to get full access to the operating system APIs. If a required signature is missing or the application is altered after signing, the **JVM** will either **refuse/restrict the API** access to the application or will fail at run-time with an error message. Attackers can obtain code-signing keys **anonymously** using prepaid credit cards and false details, sign a malicious application, and publish it on the **BlackBerry app world**. Attackers can also **compromise** a developer's system to **steal code-signing keys** and passwords to decrypt the encrypted keys.

A pictorial representation of **malicious code** signing follows:

FIGURE 16.56: Malicious Code Signing Screenshot



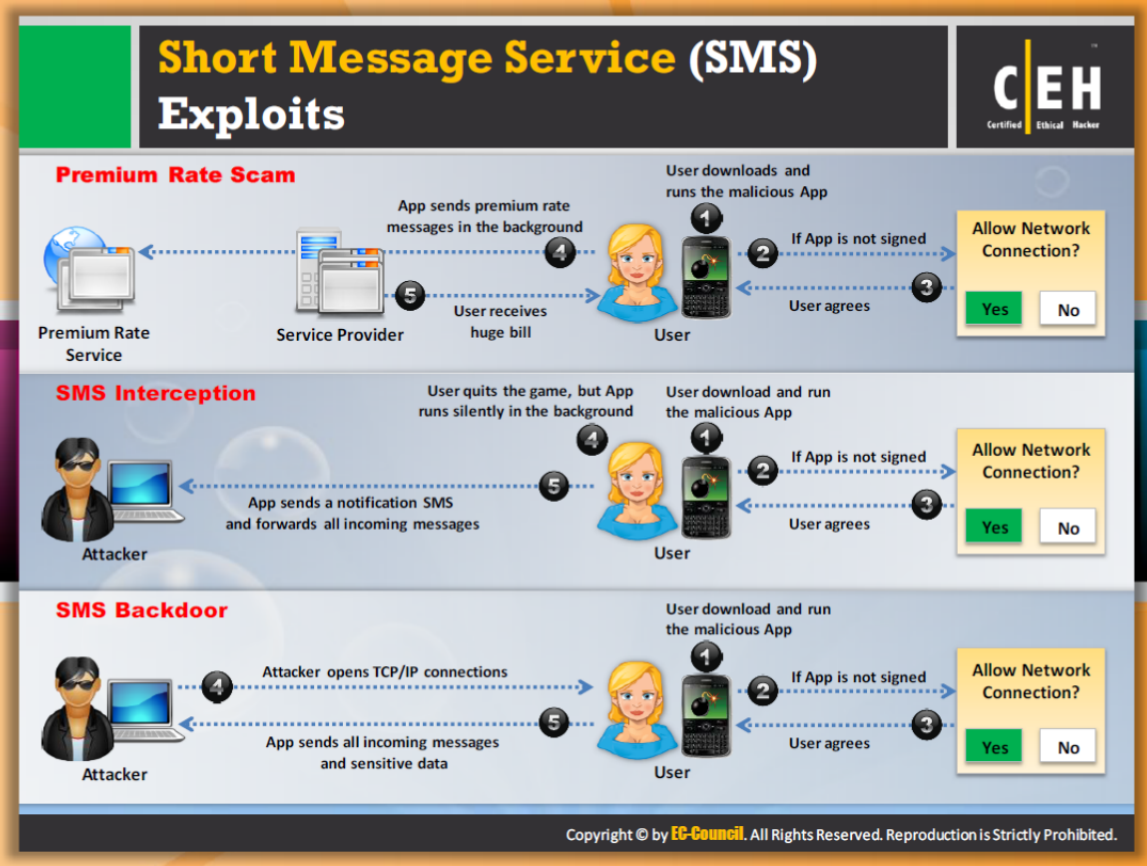
JAD File Exploits and Memory/ Processes Manipulations

JAD File Exploits

JAD (**Java Application Descriptors**) files include the attributes of a Java application, such as app description and vendor details and size, and provides the URL where the application can be downloaded. It is used as a standard way to provide **Over The Air (OTA)** installation of Java applications on **J2ME** mobile devices. Attackers can use specially crafted .jad files with spoofed information and trick users into installing malicious apps.

Memory/Processes Manipulations

Attackers can create malicious applications by creating an infinite loop, with a break condition in the middle that will always be false to bypass compiler verification. It will cause a **denial-of-service (DoS)** attack when the malicious application is run, rendering the device unresponsive.



Short Message Service (SMS) Exploits

Premium Rate Scam

Regular PC users are more likely to be targeted by premium rate "**dialers**," applications that connect a user's **modem** to a premium rate telephone number, which results in more service provider bills than expected. The same mechanism is enforced in BlackBerry but doesn't use premium rate **SMSes**.

The working of the application is illustrated in the figure that follows:

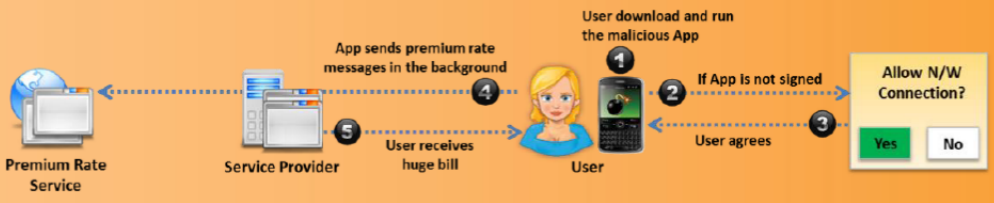


FIGURE 16.57: Short Message Service (SMS) Exploits

SMS interception

Sending and receiving of messages can be done easily by the **unsigned application**. The

messages from a **compromised** BlackBerry can be sent and received by **third parties** easily using a malicious application.

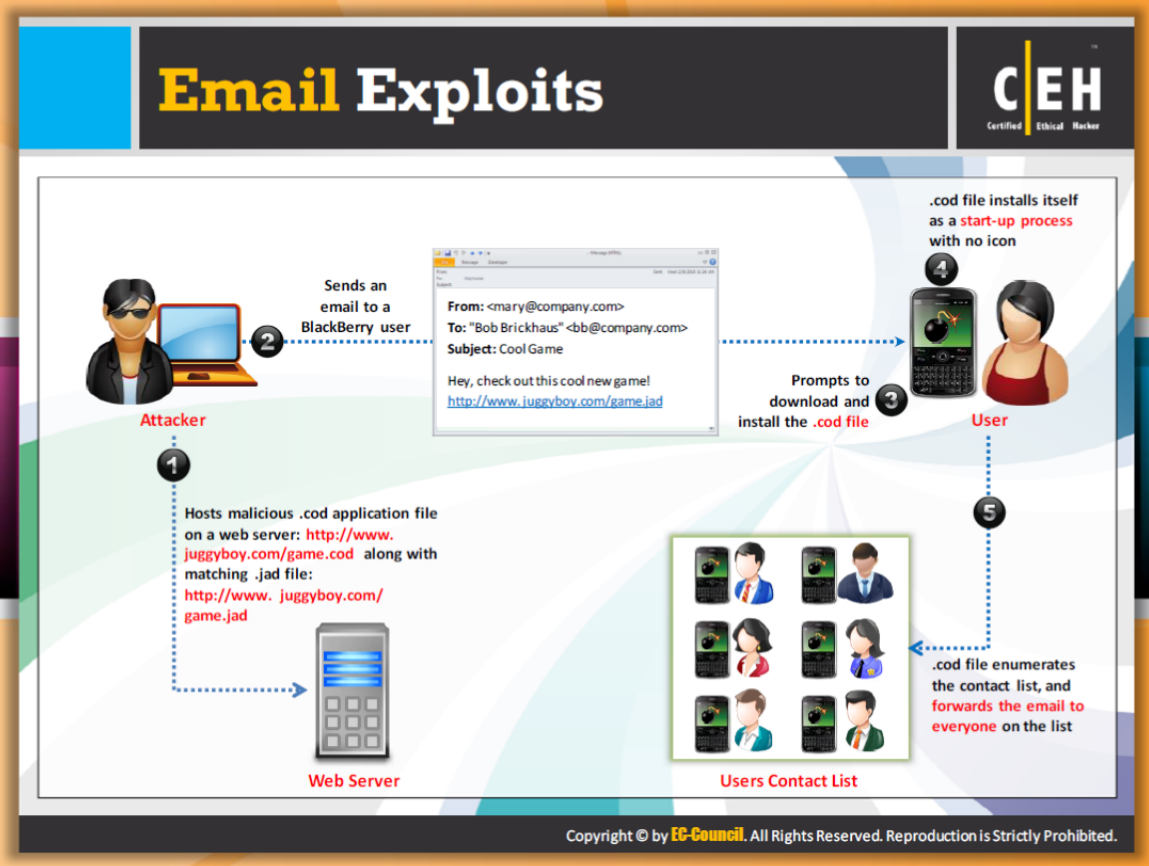
The malicious application works as shown here:

FIGURE 16.58: SMS interception

SMS Backdoor

SMS is basically used as a command and control channel by the signed malicious application for a backdoor. This malicious application has the ability to send and receive messages, steal or alter **confidential** or personal data, and open **TCP/IP connections**. The incoming SMS messages are monitored thoroughly for finding out keywords or for important phone numbers. These message are interpreted by the attacker as commands for carrying out certain malicious activities.

FIGURE 16.59: SMS Backdoor



Email Exploits

In BlackBerry mobile, all the email is sent, received, and read through the **net.rim.blackberry.api.mail** package and this package can be used only on signed applications. BlackBerry attachment service supports only files with extensions such as .doc, .pdf, .txt, .wpd, .xls, and .ppt, but it can send any kind of file via email. An **attachment** with file type .cod is not supported by BlackBerry.

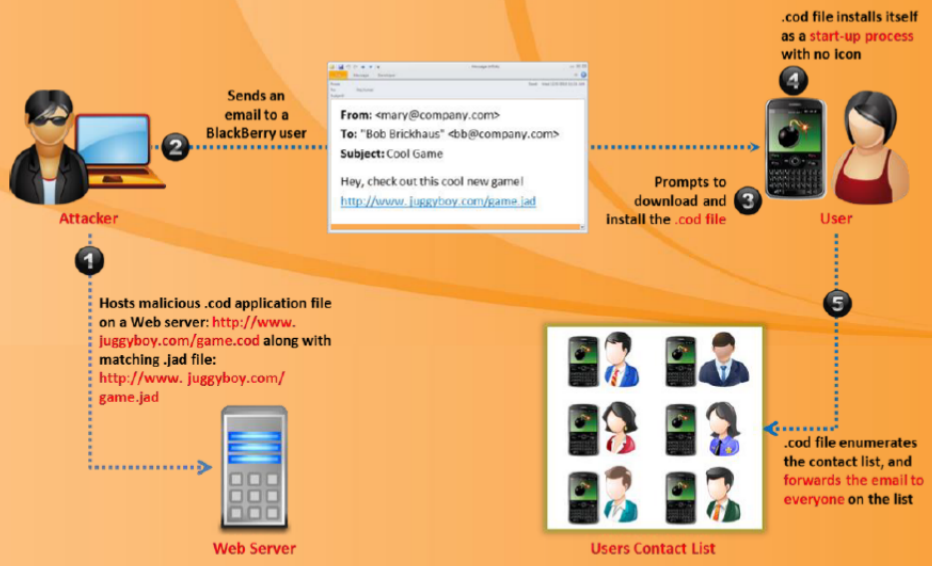
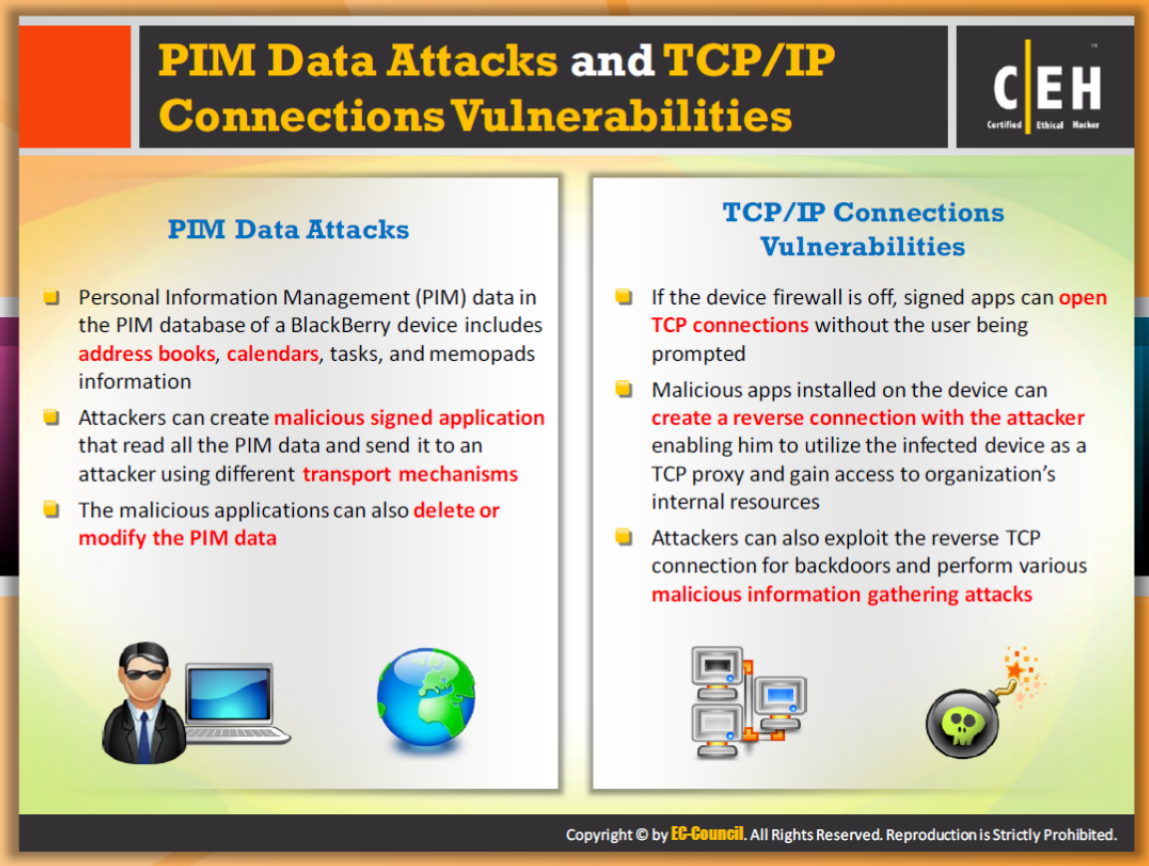


FIGURE 16.60: Email Exploits



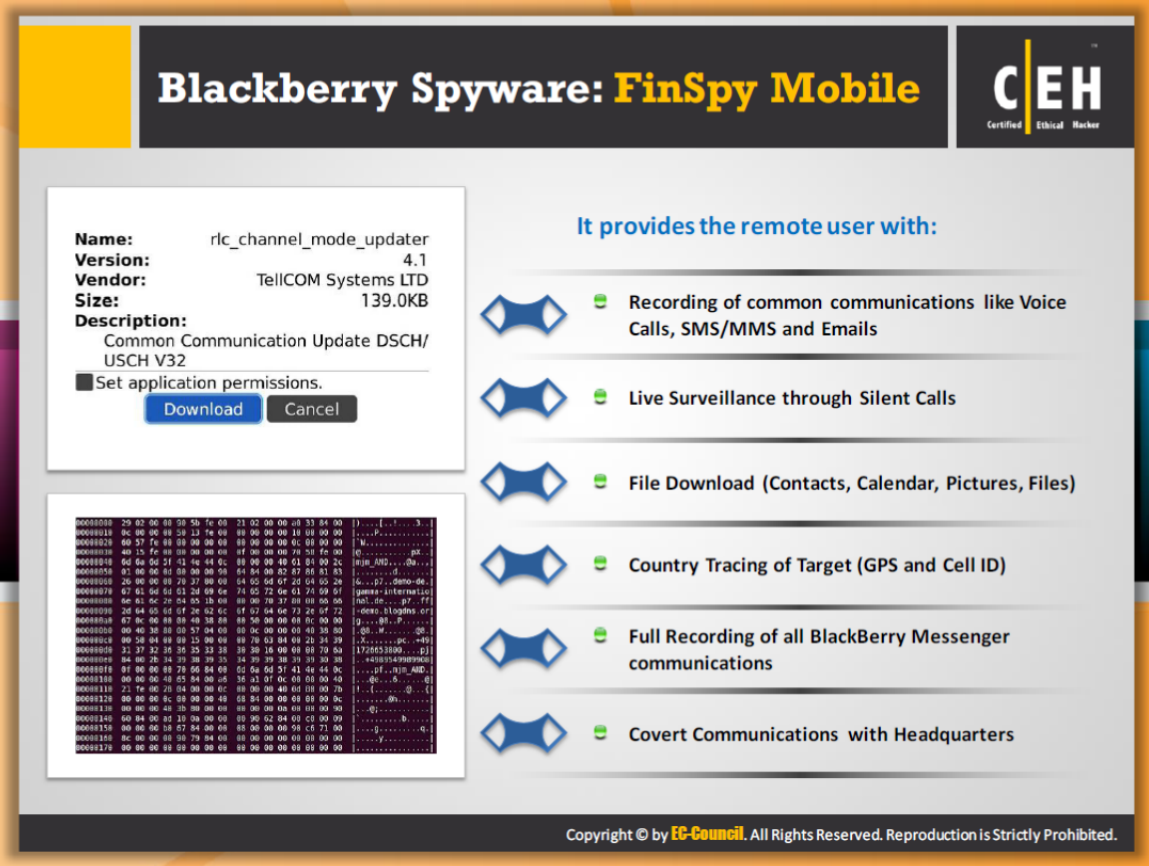
PIM Data Attacks and TCP/IP Connections Vulnerabilities

PIM Data Attacks

Personal Information Management (PIM) data in the PIM database of a BlackBerry device includes address books, calendars, tasks, and memopads information. Attackers can create malicious signed applications that read all the **PIM data** and send it to an attacker using the different **transport mechanisms**. The malicious applications can also delete or modify the PIM data.

TCP/IP Connections Vulnerabilities

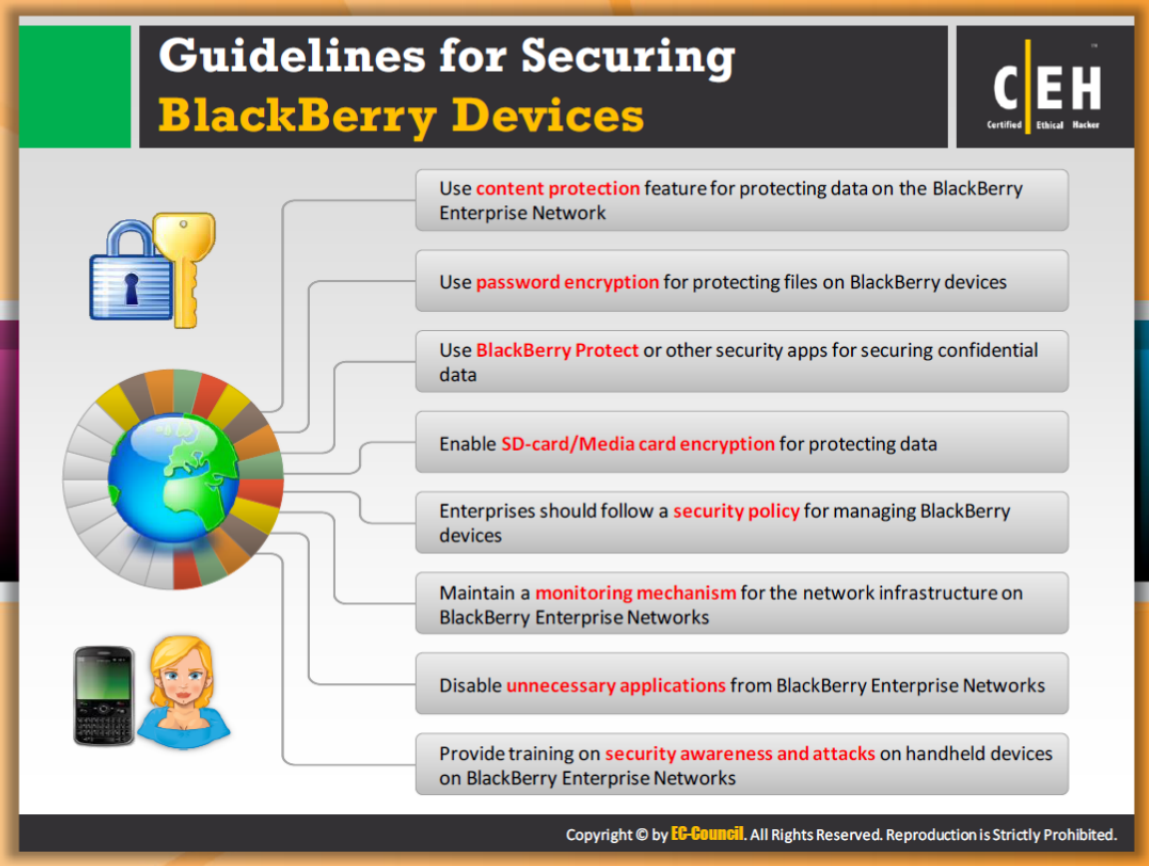
If the device firewall is off, signed apps can open **TCP connections** without the user being prompted. **Malicious apps** installed on the device can create a reverse connection with the attacker enabling him or her to utilize infected device as a **TCP proxy** and gaining access to organization's internal resources. Attackers can also exploit the reverse TCP connection for backdoors and perform various **malicious** information **gathering attacks**.



Blackberry Spyware: FinSpy Mobile

FinSpy Mobile provides the remote user with:

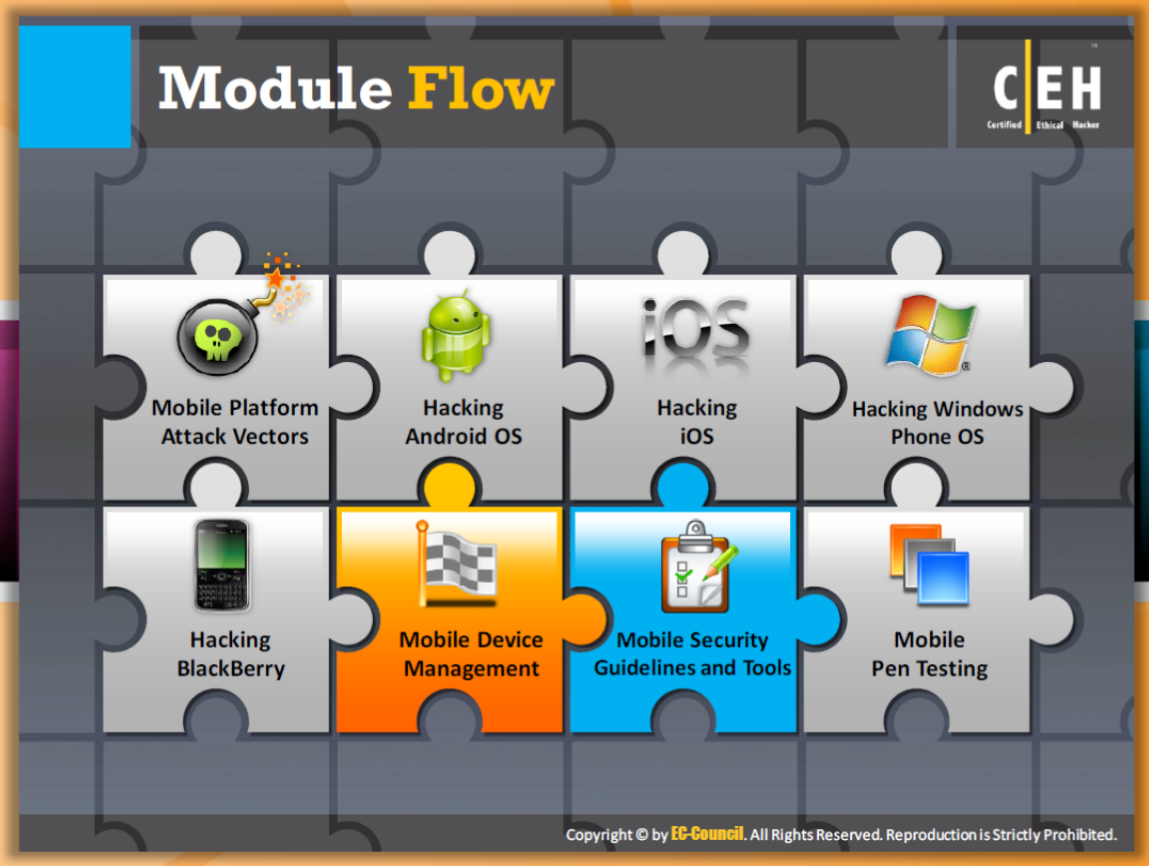
- Recording of common communications such as voice calls, SMS/MMS, and emails
- Live **surveillance** through silent calls
- File download (contacts, calendar, pictures, files)
- Country tracing of target (GPS and cell ID)
- Full recording of all BlackBerry **Messenger communications**
- Covert communications with headquarters



Guidelines for Securing BlackBerry Devices

Every user must follow guidelines to protect their BlackBerry devices against various attacks:

- Use content protection feature for **protecting** data on BlackBerry Enterprise Network
- Use password encryption for protecting files on BlackBerry devices
- Use BlackBerry Protect or other security apps for **securing confidential data**
- Enable SD-card/media card encryption for protecting data
- Enterprises should follow a **security policy** for managing BlackBerry devices
- Maintain a monitoring mechanism for network **infrastructure** on BlackBerry Enterprise Network
- Disable **unnecessary** applications from BlackBerry Enterprise Network
- Provide training on **security awareness** and attacks on handheld devices on BlackBerry Enterprise Network

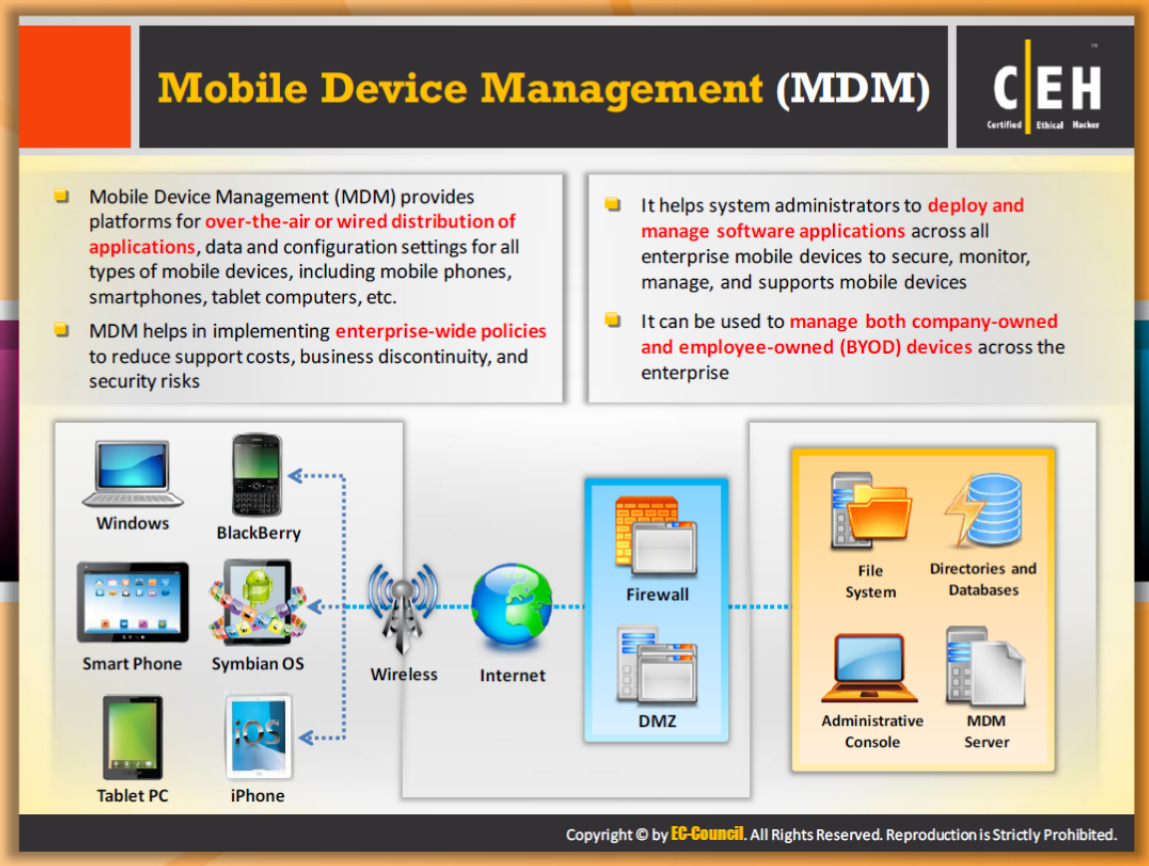


Module Flow

So far, we have discussed various mobile platform attack vectors, how to hack Android OS, iOS, Windows Phone OS, and BlackBerry. Now, we will discuss Mobile Device Management (MDM), software that secures, monitors, manages, and supports mobile devices.

Mobile Platform Attack Vectors	Hacking BlackBerry
Hacking Android iOS	Mobile Device Management
Hacking iOS	Mobile Security Guidelines and Tools
Hacking Windows Phone OS	Mobile Pen Testing

This section introduces you to MDM and its logical architecture. It also covers various MDM solutions.



Mobile Device Management (MDM)

Mobile Device Management software is a vital component that monitors, safeguards, manages, and supports different types of mobile devices and tablets including iPhone, iPad, Android, and BlackBerry, along with the applications that run on them. It **monitors** all mobile devices with different operating system such as Android, Windows, and Symbian mobile. Mobile Device Management (MDM) provides platforms for **over-the-air** or **wired distribution** of applications, data, and **configuration settings** for all types of mobile devices along with mobile phones, smartphones, tablet computers, etc. With the help of MDM, enterprise-wire policies can be implemented easily to reduce support costs, time, and business and security threats. All the company-owned, consumer-owned, as well as the **employee-owned (BYOD)** devices across the **enterprise** can be easily managed with the help of it. The MDM can reduce support cost and can minimize **business threats** just by **safeguarding** and **controlling** all the data and configuration setting of all the mobile devices in the network.

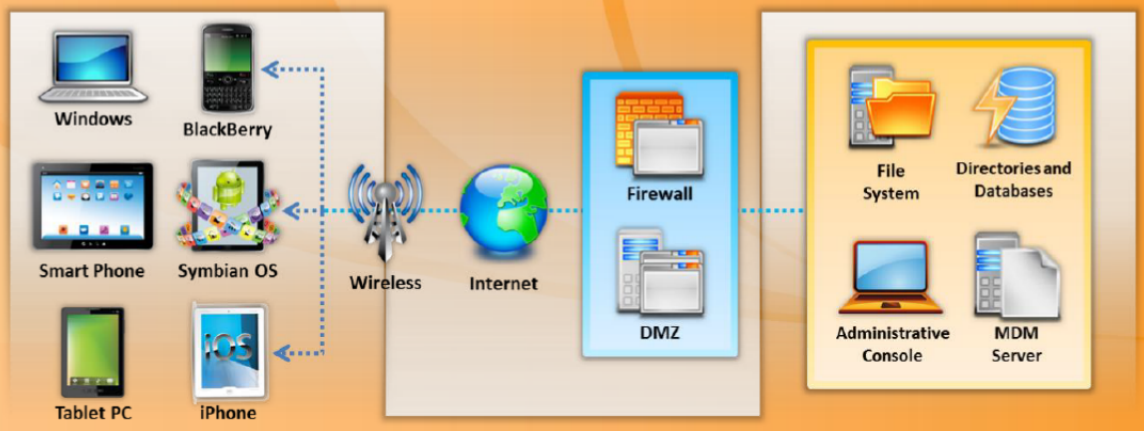
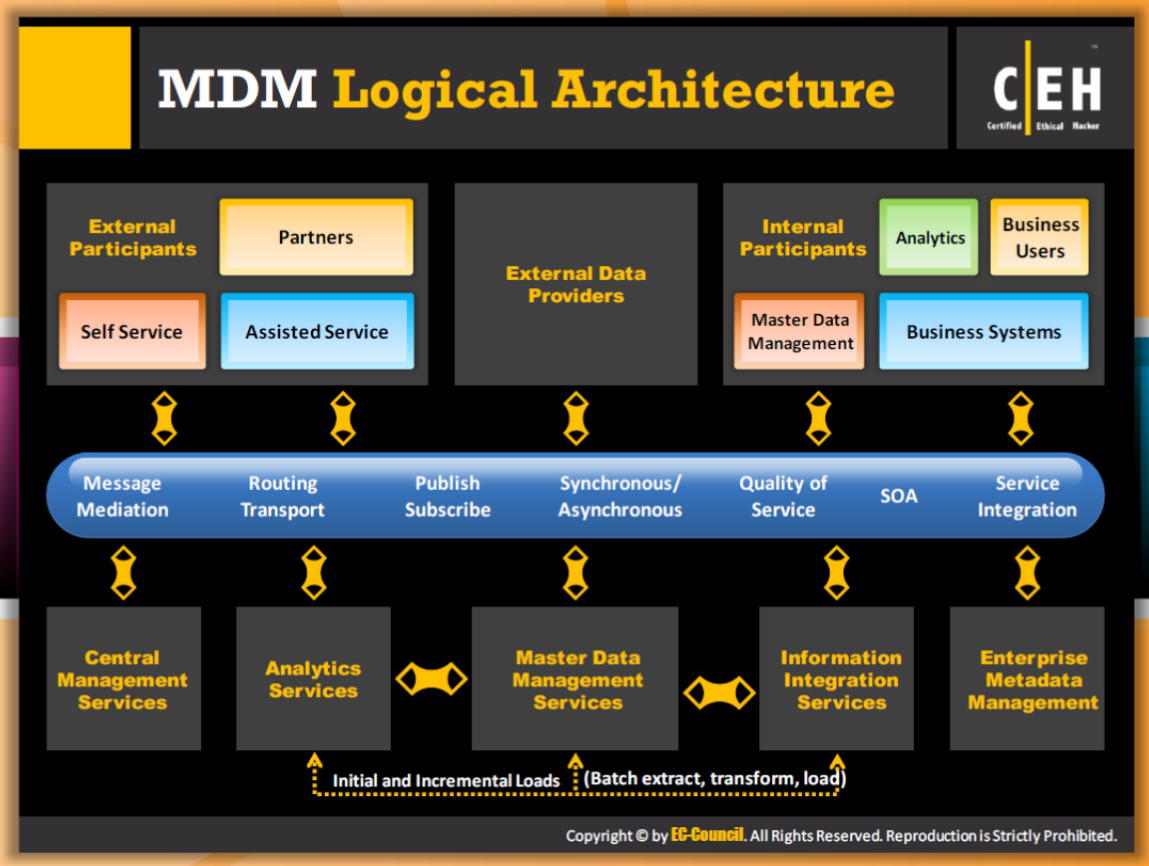


FIGURE 16.61: Mobile Device Management (MDM)



MDM Logical Architecture

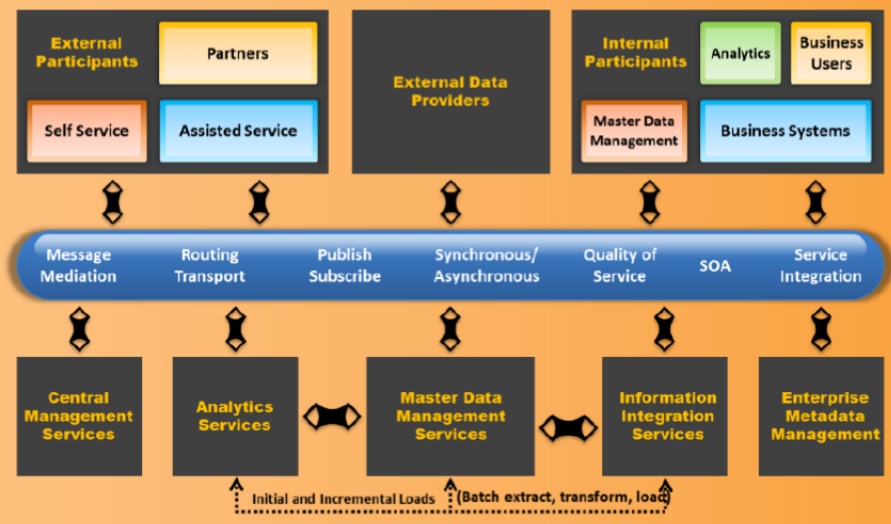


FIGURE 16.62: MDM Logical Architecture

MDM Solution: MaaS360 Mobile Device Management (MDM)

Source: http://www.maas360.com

MaaS360 Mobile Device Management (MDM) solution is a software technology that allows you to monitor and **govern mobile devices** arriving into the organization, whether they are provided by the company or part of a **Bring Your Own Device** (BYOD) program. This **technique** allows organizations to implement the MDM lifecycle for devices such as smartphones and tablets including iPhones, iPads, Androids, Windows Phones, BlackBerrys, and Kindle Fires. Using the integrated cloud platform, the **MaaS360 streamlines MDM** with **improved visibility** and **control** that spans across mobile devices, applications, and documents.

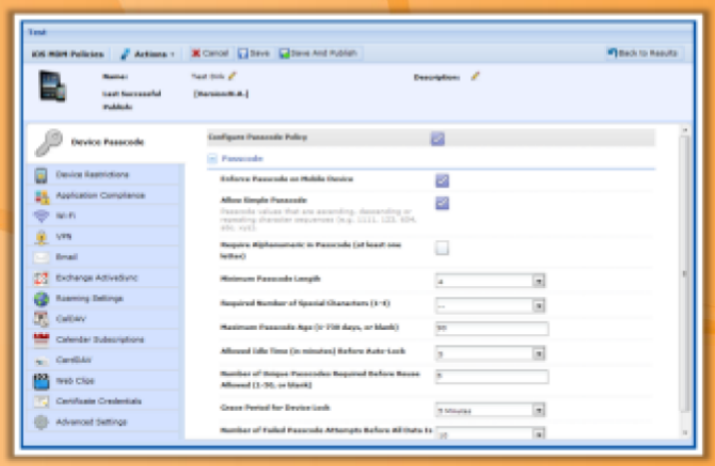
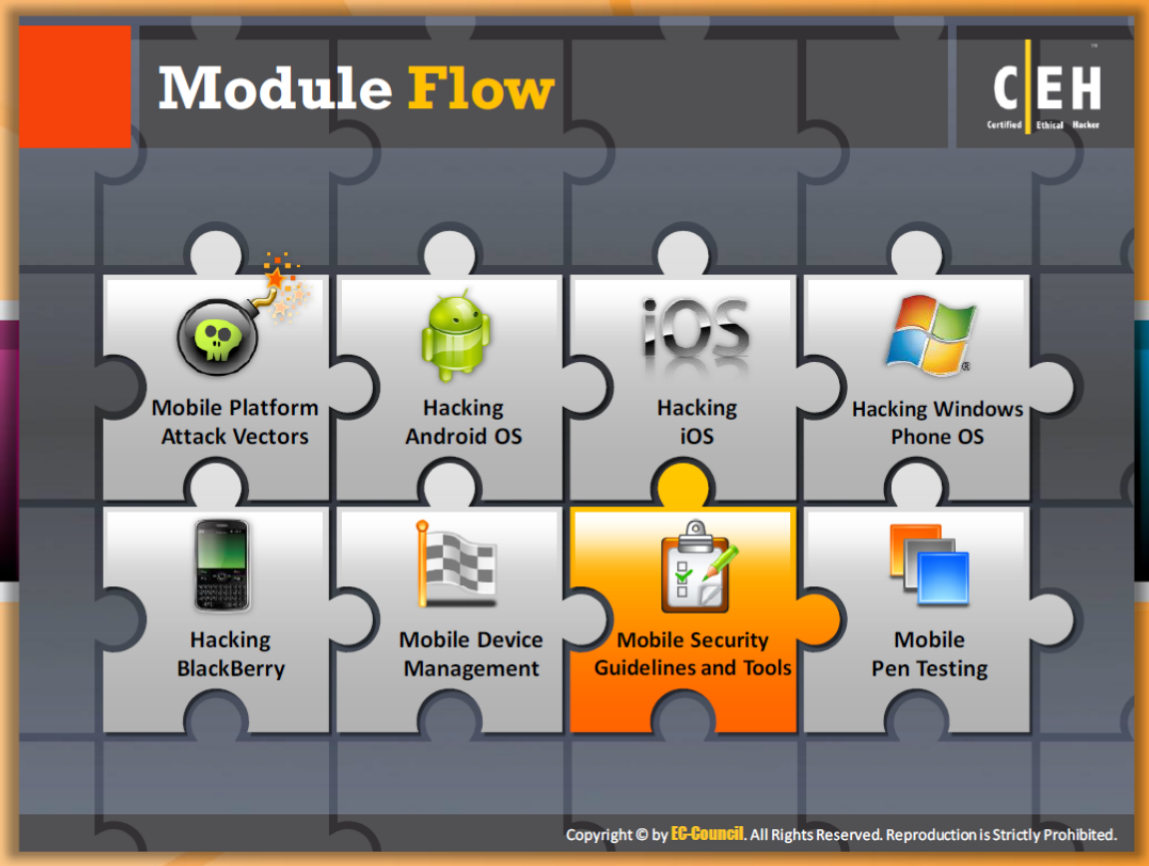


FIGURE 16.63: MaaS360 Mobile Device Management (MDM)

MDM Solutions

In addition to MaaS360 Mobile Device Management (MDM), software technologies that offer integrated mechanisms of all mobile devices in an organization for MDM include:

- Citrix XenMobile MDM available at http://www.zenprise.com
- Absolute Manage MDM available at http://www.absolute.com
- SAP Afaria available at http://www.sybase.com
- Device Management Centre available at http://www.sicap.com
- AirWatch available at http://www.air-watch.com
- Good Mobile Manager available at http://www1.good.com
- MobileIron available at http://www.mobileiron.com
- Rule Mobility available at http://www.tangoe.com
- TARMAC available at http://www.tarmac-mdm.com
- MediaContact available at http://www.device-management-software.com

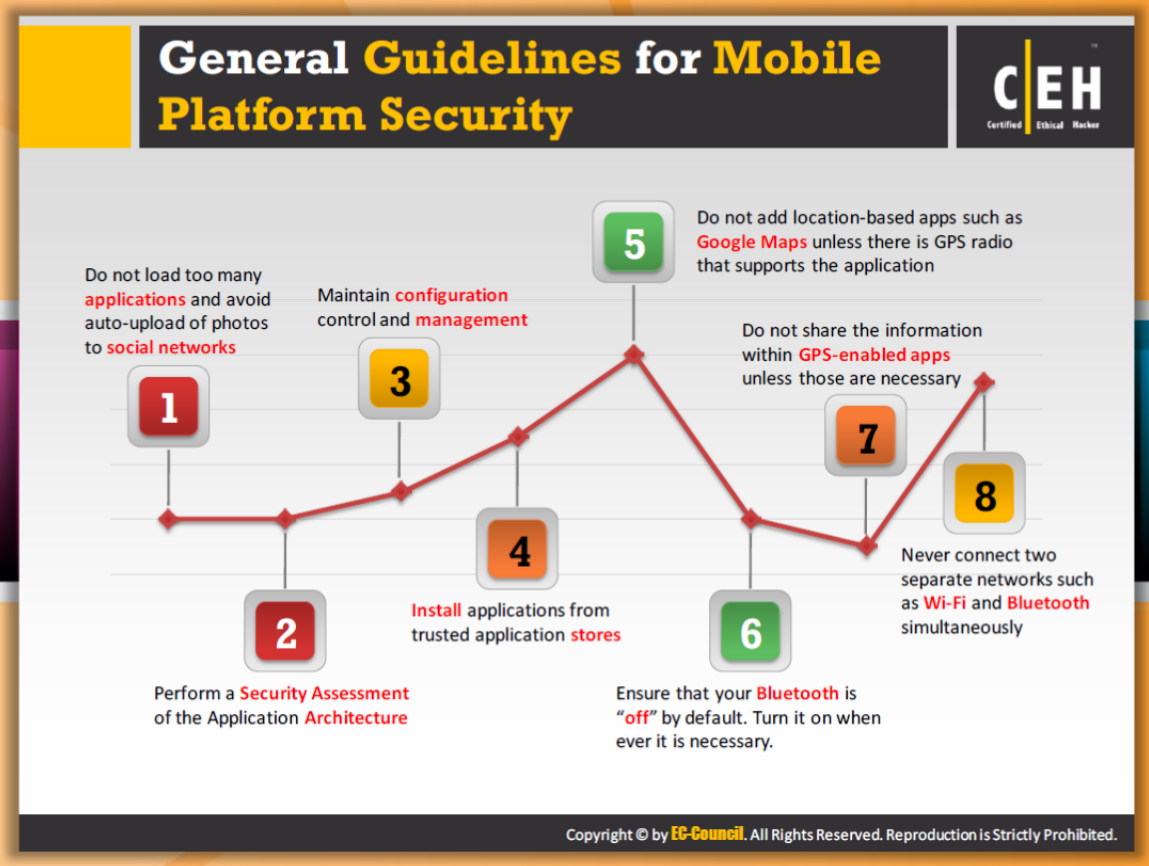


Module Flow

So far, we have discussed various topics such as mobile platform attack vectors, hacking methods of Android OS, iOS, Windows Phone OS, BlackBerry, and how to manage mobile devices. All these topics discussed so far help in testing mobile devices. Now, we will discuss mobile security guidelines and tools that help in securing the mobile devices.

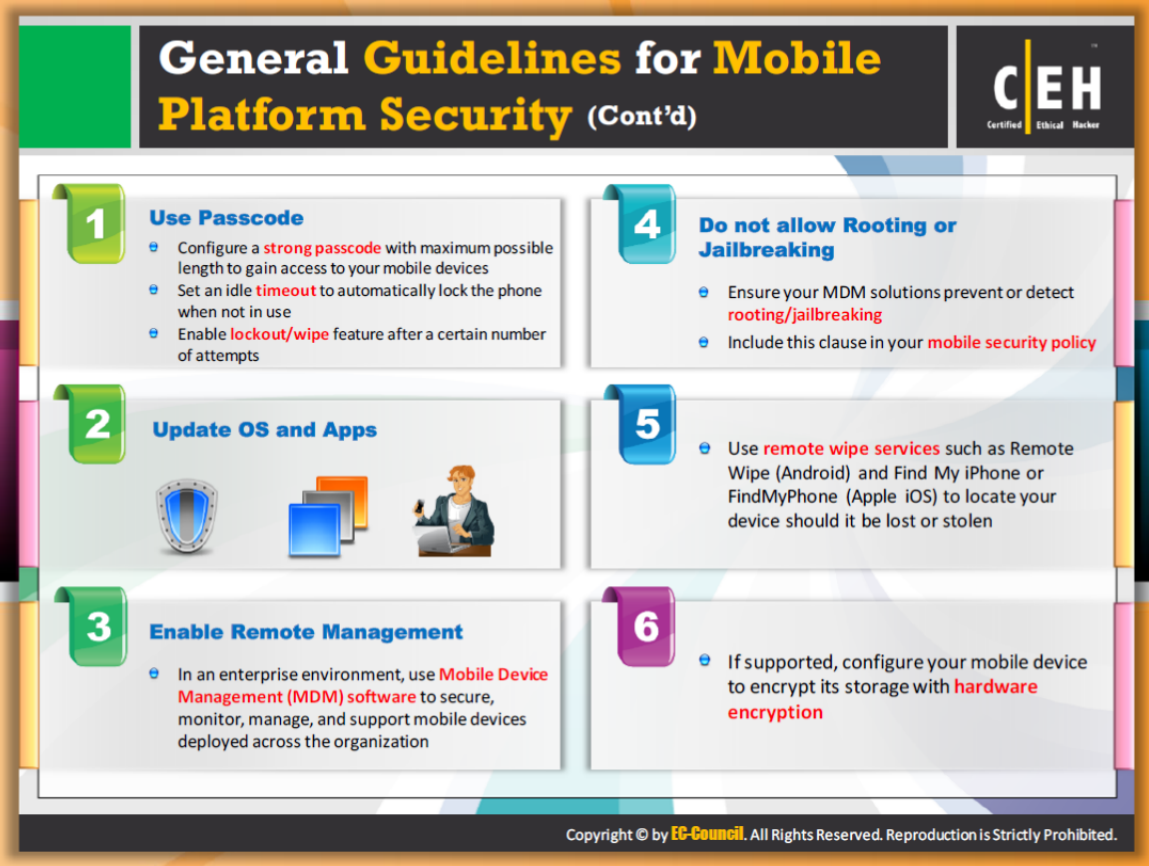
Mobile Platform Attack Vectors	Hacking BlackBerry
Hacking Android iOS	Mobile Device Management
Hacking iOS	Mobile Security Guidelines and Tools
Hacking Windows Phone OS	Mobile Pen Testing

This section is dedicated to mobile security guidelines.



General Guidelines for Mobile Platform Security

- Do not load too many applications and avoid **auto-upload** of photos to social networks
- Perform a security assessment of the application **architecture**
- Maintain **configuration control** and management
- Install applications from trusted application stores
- Do not add **location-based features** such as Google Maps unless there is a component that supports the application
- Ensure that your Bluetooth is "off" by default; turn it on when ever it is necessary
- Do not share information within **GPS-enabled apps** unless necessary
- Never connect two separate networks such as **Wi-Fi and Bluetooth** simultaneously

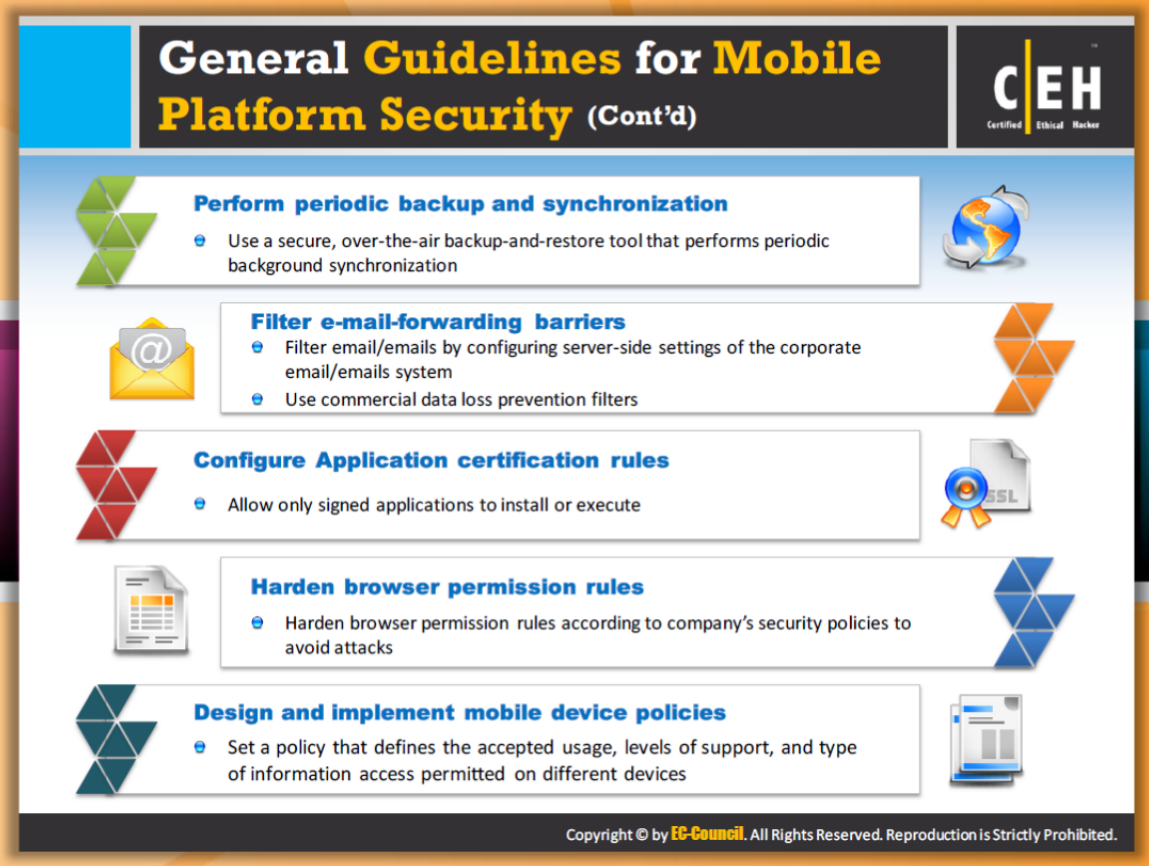


General Guidelines for Mobile Platform Security (Cont'd)

The following guidelines will help you to secure your mobile device from many type of attack:

1. **Use a passcode for mobile device security**
   - Configure a **strong passcode** with maximum possible length to gain access to your mobile devices
   - Set an idle timeout to **automatically lock** the phone when not in use
   - Enable **lockout/wipe** feature after a certain number of attempts
2. **Update OS and apps regularly**
3. **Enable Remote Management**
   - In an enterprise environment, use Mobile Device Management (MDM) software to secure, monitor, manage, and support mobile devices deployed across the organization
4. **Do not allow rooting or jailbreaking**
   - Ensure your MDM solutions **prevent or detect rooting/jailbreaking**
   - Include this clause in your **mobile security policy**

5. Use remote wipe services such as Remote Wipe (Android) and Find My iPhone or FindMyPhone (Apple iOS) to locate your device should it be lost or stolen
6. If supported, configure your mobile device to encrypt its storage with hardware encryption



General Guidelines for Mobile Platform Security (Cont'd)

Perform periodic backup and synchronization

- Use a secure, over-the-air backup-and-restore tool that performs periodic background synchronization

Filter email-forwarding barriers

- Filter emails by configuring server-side settings of the corporate email system
- Use commercial data loss prevention filters

Configure application certification rules

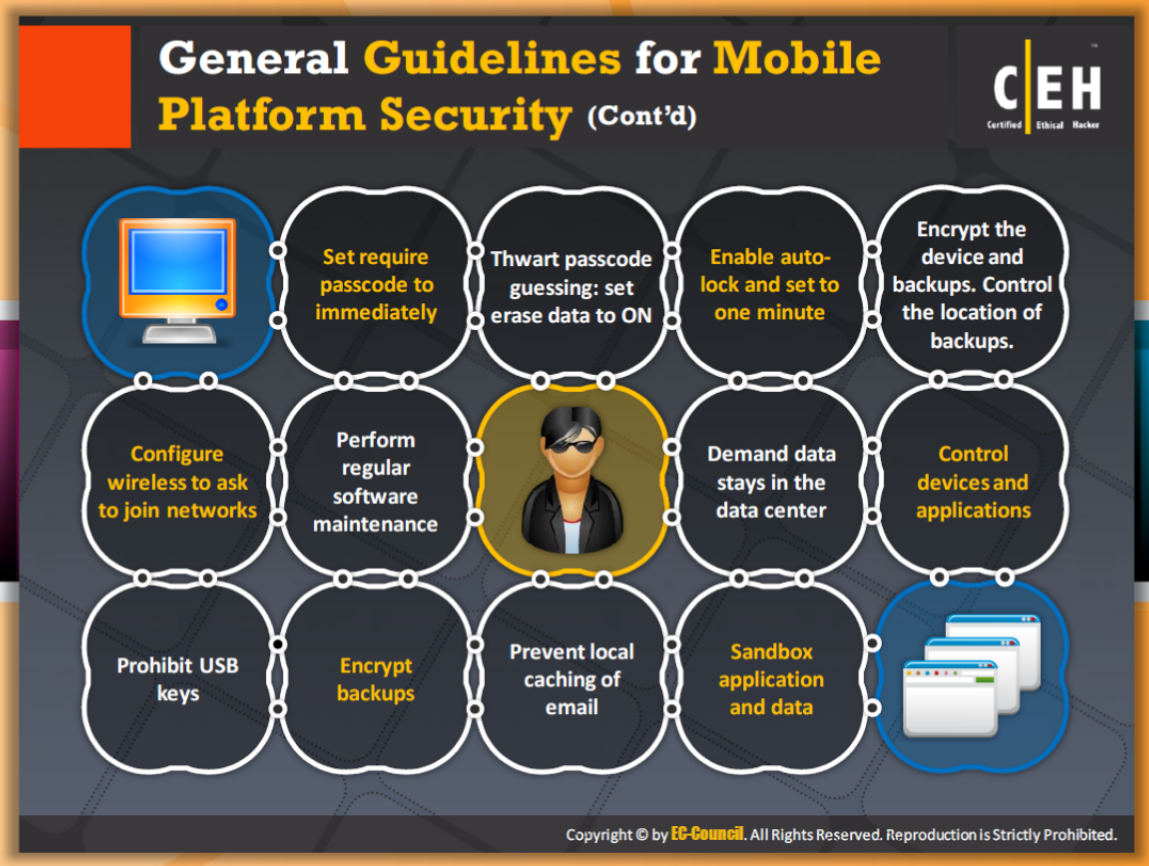
- Allow only signed applications to install or execute

Harden browser permission rules

- Harden browser permission rules according to company's security policies to avoid attacks

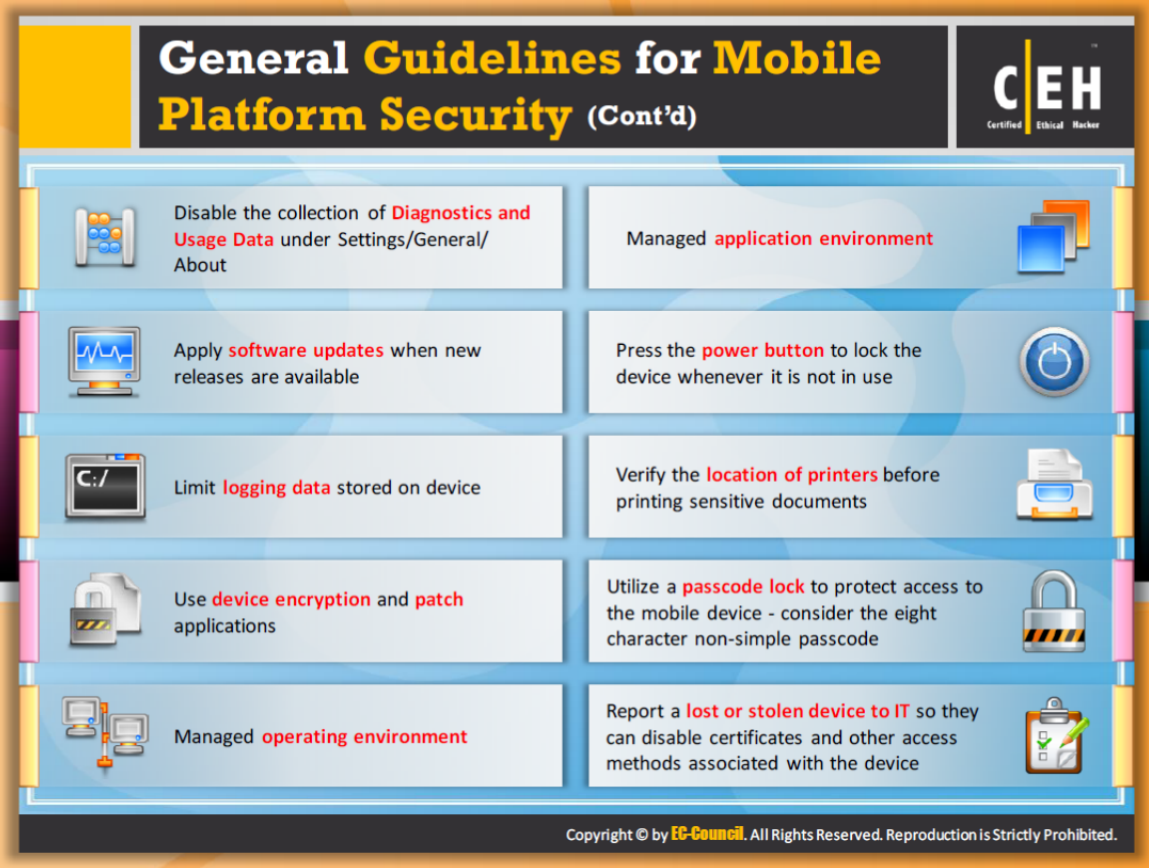
Design and implement mobile device policies

- Set a policy that defines the accepted usage, levels of support, type of information access on different devices



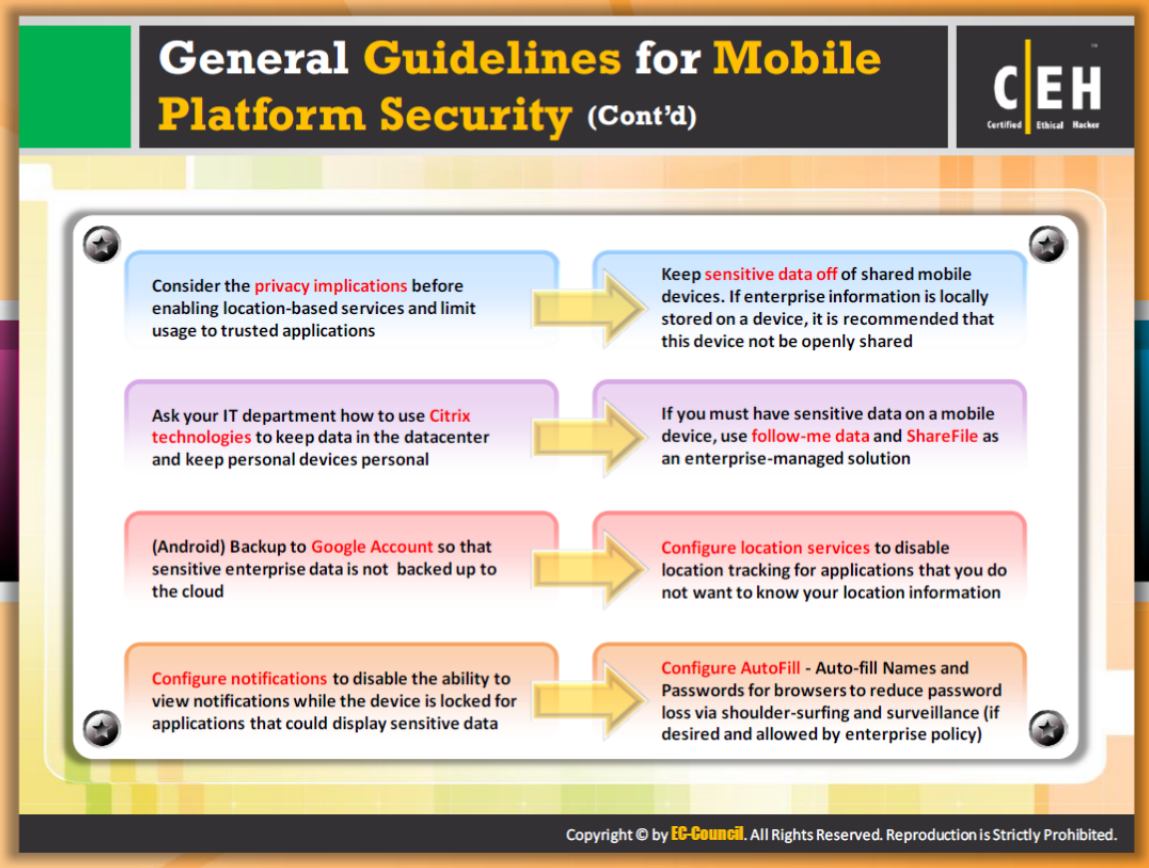
General Guidelines for Mobile Platform Security (cont'd)

- Set Require **Passcode** to Immediately
- Thwart passcode guessing: Set Erase Data to ON
- Enable **Auto-Lock** and set to one minute
- Encrypt the device and backups
- Control the location of backups
- **Configure** wireless to Ask to Join Networks
- Software **maintenance**
- Data stays in the data center
- App/device **control**
- No USB key capability
- **Encrypted backups**
- Email not cached locally
- Application/data sandboxing



General Guidelines for Mobile Platform Security (Cont'd)

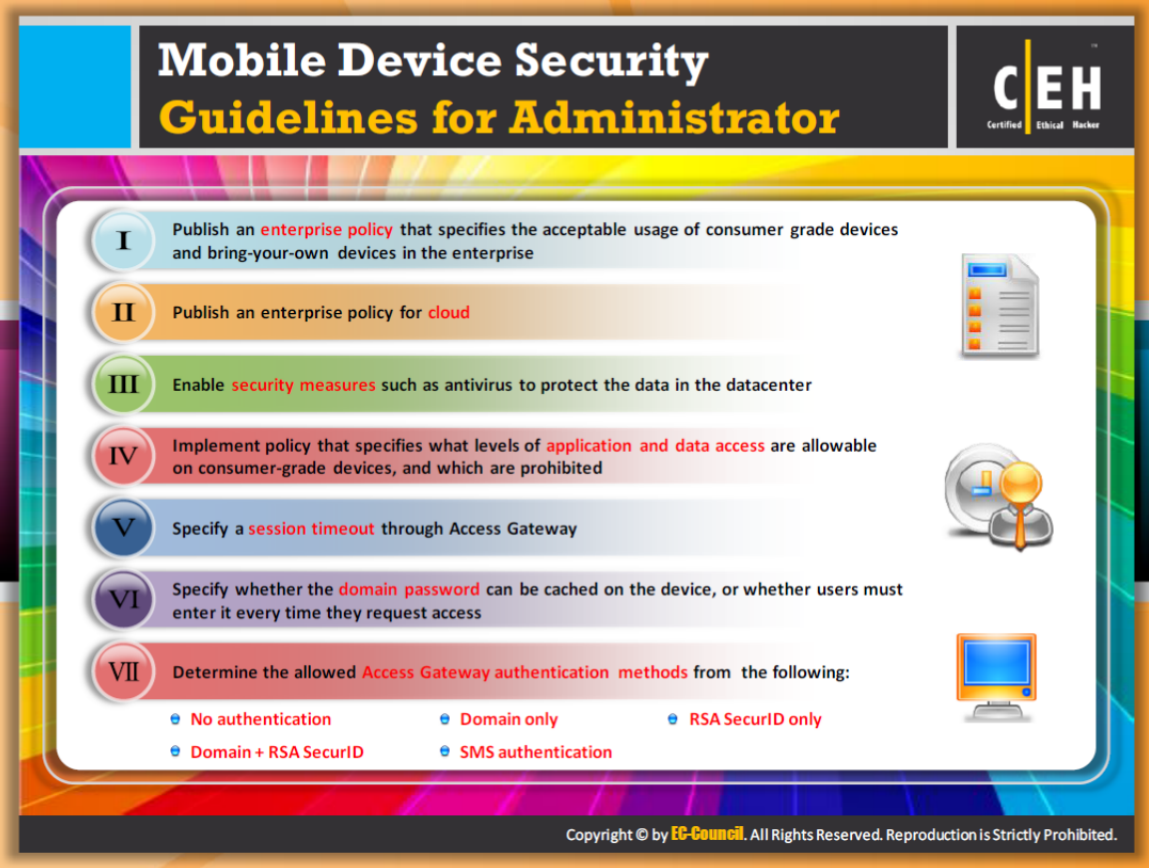
- **Disable** the collection of **Diagnostics** and **Usage Data** under Settings/General/About
- Apply software **updates** when new releases are available
- Logging and limited data on device
- Device **encryption** and application patching
- Managed operating environment
- Managed application environment
- Press the power button to **lock the device** whenever it is not in use
- **Verify** the location of printers before printing **sensitive** documents
- Utilize a passcode lock to protect access to the mobile device; consider the eight character non-simple **passcode**
- Report a lost or stolen device to IT so they can disable certificates and other access methods associated with the device



General Guidelines for Mobile Platform Security (Cont'd)

- Consider the privacy implications before enabling **location-based** services and limit usage to trusted applications
- Ask your IT department how to use **Citrix technologies** to keep data in the datacenter and keep personal devices personal
- (Android) **Backup to Google Account** so that sensitive enterprise data is not backed up to the cloud
- **Configure notifications** to disable the ability to **view notifications** while the device is locked for applications that could display sensitive data
- Keep **sensitive data** off of shared mobile devices. If enterprise information is locally stored on a device, it is recommended that this device not be openly shared
- If you must have sensitive data on a mobile device, use follow-me data and ShareFile as an enterprise-managed solution
- **Configure location** services to disable location tracking for applications that you do not want to know your location information

- **Configure AutoFill**; Auto-fill Names and Passwords for browsers to reduce password loss via shoulder-surfing and surveillance (if desired and allowed by enterprise policy)



Mobile Device Security Guidelines for Administrator

The administrator should follow the guidelines listed here to implement mobile device security:

1. Publish an **enterprise policy** that specifies the acceptable usage of consumer grade devices and bring-your-own devices in the enterprise
2. Publish an enterprise policy for cloud
3. Enable **security measures** such as antivirus to protect the data in the datacenter
4. Implement policy that specifies what levels of application and data access are allowable on **consumer-grade devices**, and which are **prohibited**
5. Specify a session timeout through **Access Gateway**
6. Specify whether the domain password can be cached on the device, or whether users must enter it every time they **request access**
7. Determine the allowed **Access Gateway** authentication methods from the following:
   - No authentication
   - Domain only

- RSA SecurID only
- Domain + RSA SecurID
- SMS authentication

Mobile Protection Tool: BullGuard Mobile Security

Source: http://www.bullguard.com

BullGuard Mobile Security delivers complete mobile phone antivirus against all mobile phone viruses. It tracks a stolen or lost mobile via the **built-in GPS**, locks it, or wipe the data off it, to make sure no one can access your personal information, passwords, and financial data.

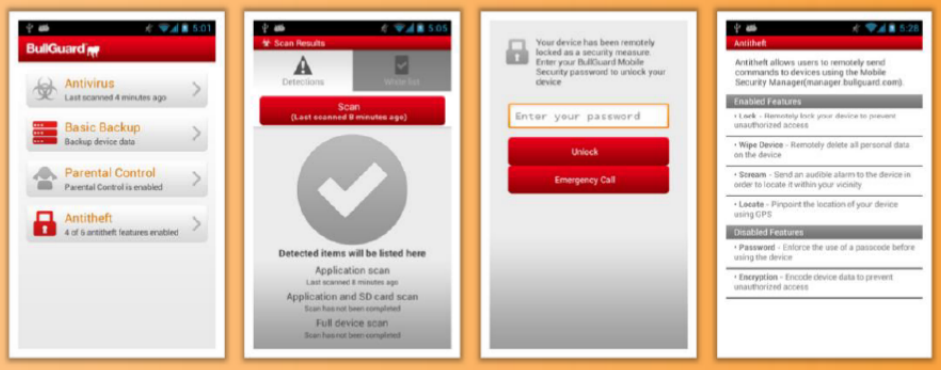


FIGURE 16.64: BullGuard Mobile Security Screenshot

Mobile Protection Tool: Lookout

Source: https://www.lookout.com

Lookout is a mobile protection tool that allows you to protect your mobile from **mobile threats**. It helps you to avoid risky behavior such as connecting to an unsecured Wi-Fi network, **downloading a malicious app** or clicking on a **fraudulent link** in order to prevent identity theft, financial fraud and the loss of your most **personal data**. This provides safe, secure, and seamless backup of your mobile data, automatically over the air, and allows you to find your phone if it's lost or stolen. The dashboard allows you to remotely manage your phone.

FIGURE 16.65: Lookout Screenshot

Mobile Protection Tool: WISeID

Source: http://www.wiseid.mobi

WISeID provides secure and **easy-to-use encrypted storage** for **personal data**, personal identifiable information (PII), PINs, credit and loyalty cards, notes, and other information. WISeID allows you to store your websites, user names, and passwords and quickly log on to your favorite websites through your mobile device.

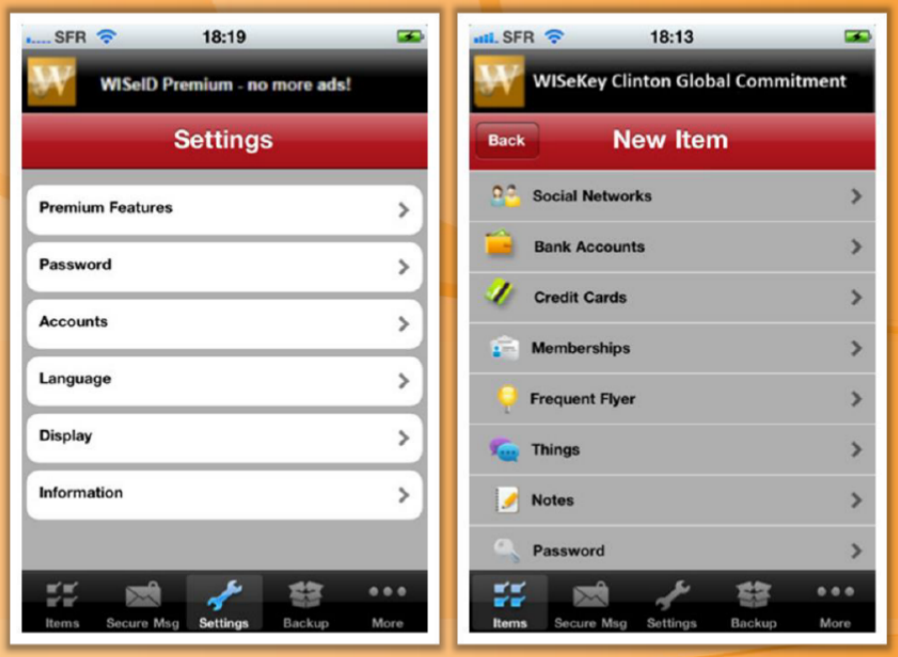
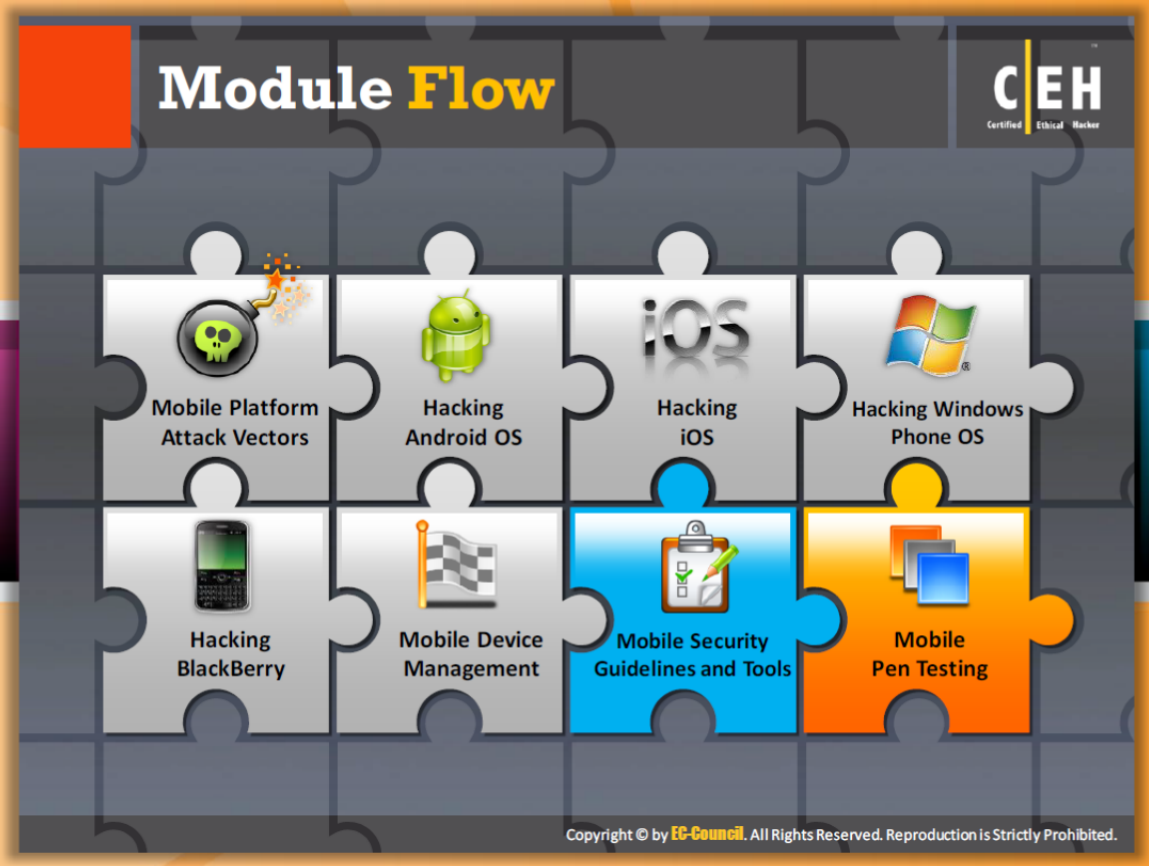


FIGURE 16.66: WISelD Screenshot

Mobile Protection Tools

In addition to the tools including **BullGuard Mobile Security**, Lookout and WISeID, there are a number of other tools available for mobile protection:

- McAfee Mobile Security available at https://www.mcafeemobilesecurity.com
- AVG AntiVirus Pro for Android available at http://www.avg.com
- avast! Mobile Security available at http://www.avast.com
- Norton Mobile Security available at http://us.norton.com
- ESET Mobile Security available at http://www.eset.com
- Kaspersky Mobile Security available at http://www.kaspersky.com
- F-Secure Mobile Security available at http://www.f-secure.com
- Trend Micro Mobile Security available at http://www.trendmicro.com
- Webroot Secure Anywhere Mobile available at http://www.webroot.com
- NetQin Mobile Security available at http://www.netqin.com

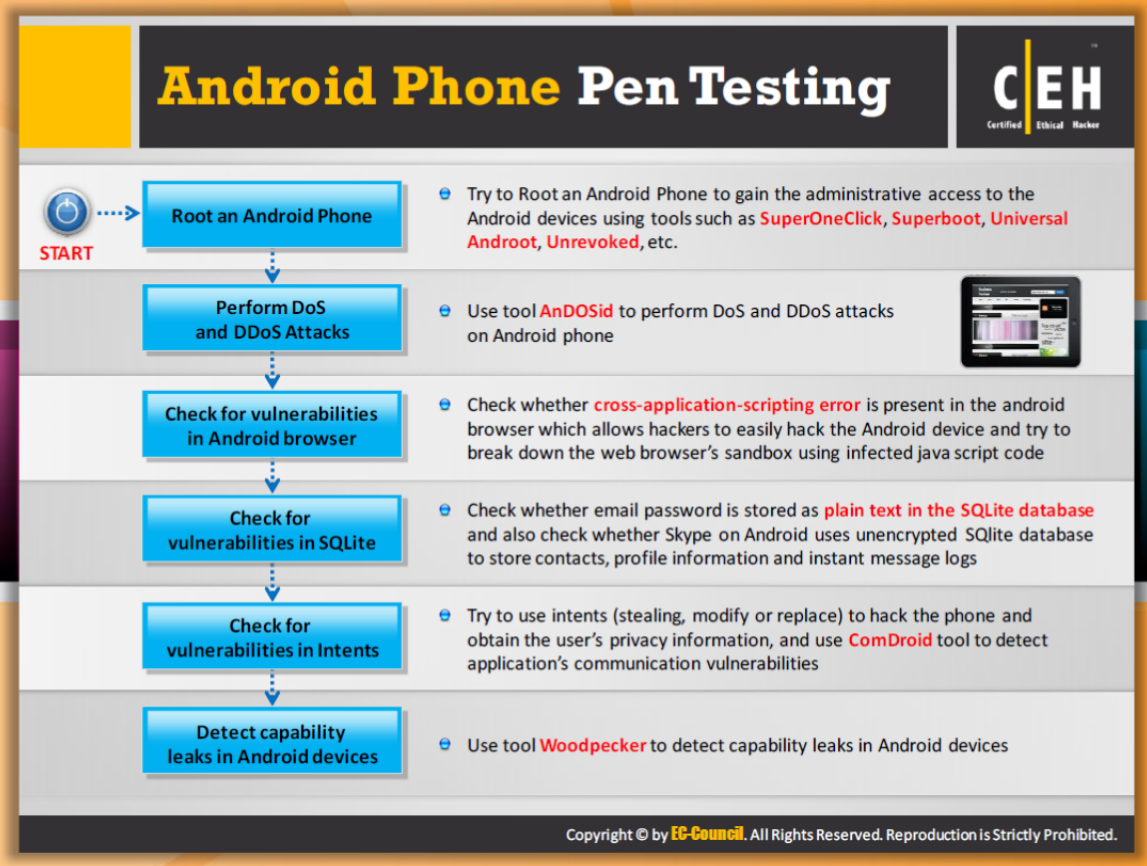


Module Flow

With the increasing use of smartphones for business and online transactions, attackers are concentrating on launching various kinds of attacks for financial gain. Therefore, as a smart mobile phone user, you should check your mobile security against possible attacks. You can test the security with the help of mobile pen testing.

Mobile Platform Attack Vectors	Hacking BlackBerry
Hacking Android iOS	Mobile Device Management
Hacking iOS	Mobile Security Guidelines and Tools
Hacking Windows Phone OS	Mobile Pen Testing

This section describes the step-by-step process of mobile pen testing.



Android Phone Pen Testing

The security testing differs based on the mobile operating system or **mechanism**. Let's begin with Android phone **pen testing**. The steps involved in Android phone pen testing are:

Step 1: Root an Android phone

Try to root an Android phone to gain the **administrative access** to the Android devices using tools such as SuperOneClick, Superboot, Universal Androot, Unrevoked, etc.

Step 2: Perform DoS and DDoS attacks

Use tool **AnDOSid** to perform DoS and DDoS attacks on the Android phone.

Step 3: Check for vulnerabilities in the Android browser

Check whether **cross-application-scripting error** is present in the Android browser, which allows hackers to easily hack the Android device and try to break down the web browser's sandbox using infected **JavaScript code**

Step 4: Check for vulnerabilities in SQLite

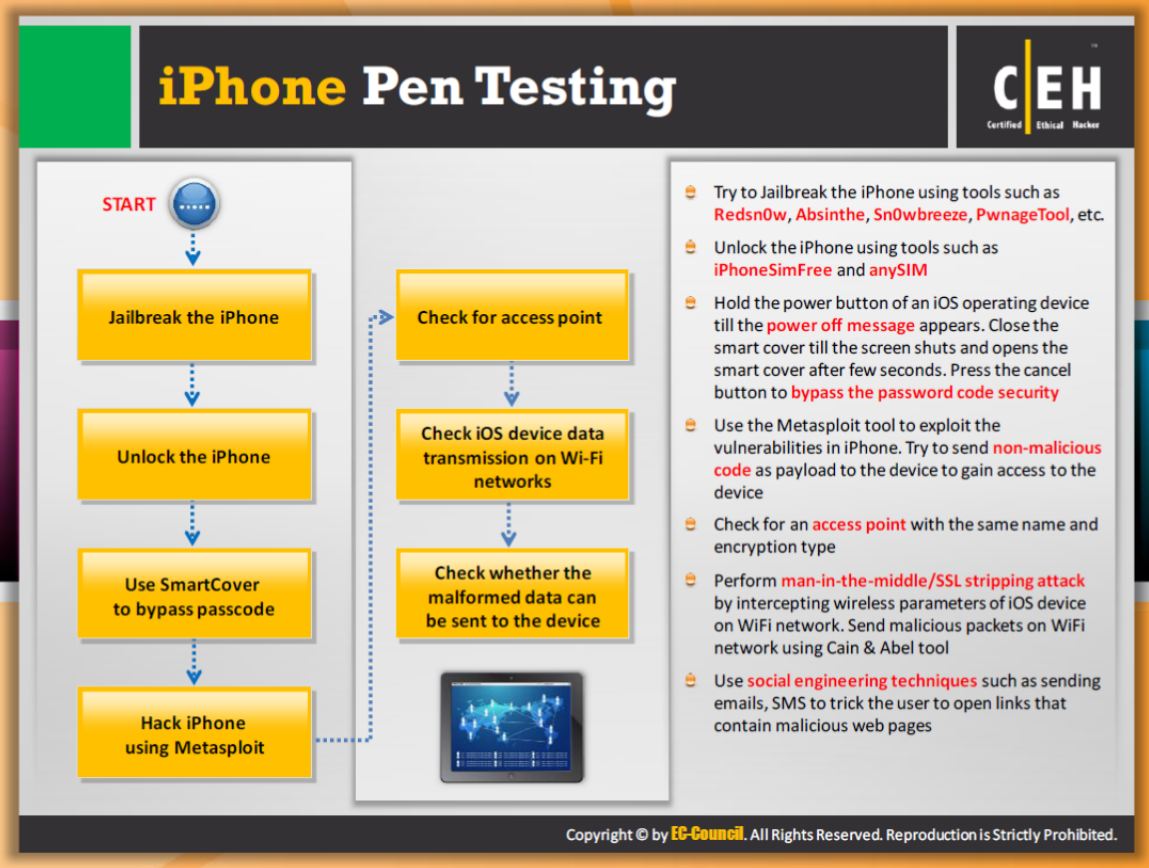
Check whether an email password is stored as plaintext in the SQLite database and also check whether Skype on Android uses an **unencrypted** SQlite database to store contacts, profile information, and instant message logs

Step 5: Check for vulnerabilities in Intents

Try to use intents (steal, modify, or replace) to hack the phone and obtain the user's privacy information and use **ComDroid tool** to detect application's communication vulnerabilities

Step 6: Detect capability leaks in Android devices

Use tool Woodpecker to **detect capability leaks** in Android devices.



iPhone Pen Testing

In order to test your iPhone for **Potential vulnerabilities**, follow the steps here:

Step 1: Jailbreak the iPhone

Try to jailbreak the iPhone using tools such as Redsn0w, Absinthe, Sn0wbreeze, PwnageTool, etc.

Step 2: Unlock the iPhone

Unlock the iPhone using tools such as iPhoneSimFree and anySIM.

Step 3: Use SmartCover to bypass passcode

Hold the power button of an **iOS operating device** until the power off message appears. Close the smart cover until the screen shuts and opens the smart cover after few seconds. Press the cancel button to bypass the password **code security**.

Step 4: Hack iPhone using Metasploit

Use the Metasploit tool to exploit the **vulnerabilities** in the iPhone. Try to send non-malicious code as payload to the device to gain access to the device.

Step 5: Check for Access Point

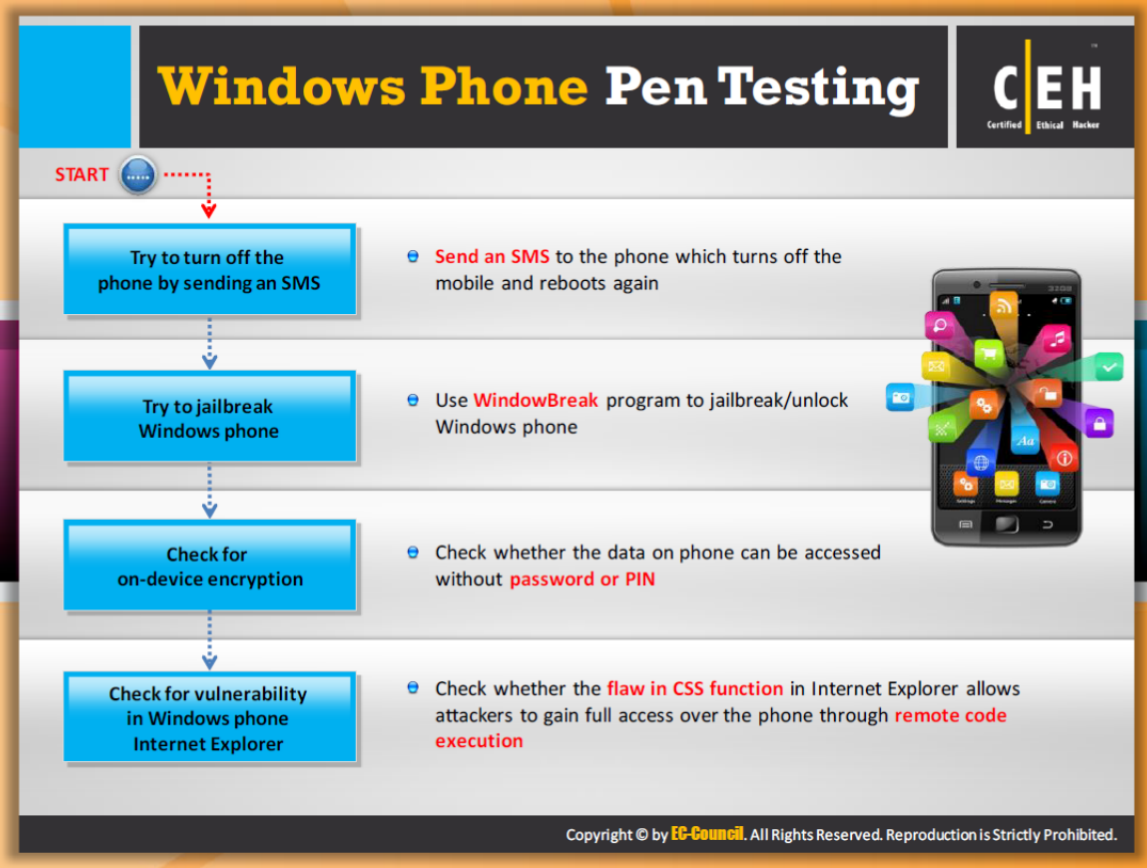
Check for access point with the same name and **encryption type**.

Step 6: Check iOS device data transmission on Wi-Fi networks

Perform a man-in-the-middle/SSL stripping attack by intercepting wireless parameters of iOS device on a Wi-Fi network. Send **malicious packets** on the Wi-Fi network using the Cain & Abel tool.

Step 7: Check whether the malformed data can be sent to the device

Use social engineering **techniques** such as sending emails or SMS to trick the user into opening links that contain malicious web pages.



Windows Phone Pen Testing

You can test a Windows phone for **security flaws** by following the Windows phone pen testing steps mentioned here:

Step 1: Try to turn off the phone by sending an SMS

Send an SMS to the phone, which turns off the mobile and **reboots** it again.

Step 2: Try to jailbreak the Windows phone

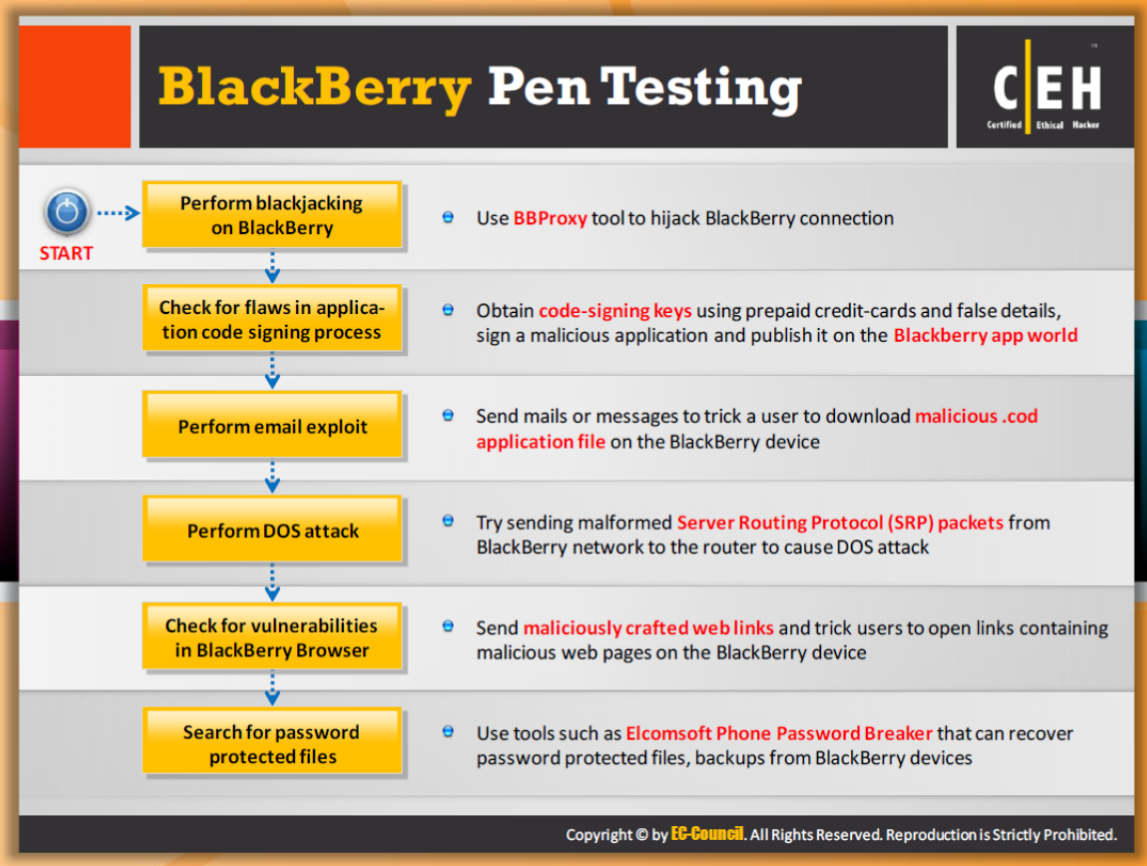
Use the WindowBreak program to **jailbreak/unlock** the Windows phone.

Step 3: Check for on-device encryption

Check whether the data on the phone can be **accessed** without a password or PIN.

Step 4: Check for a vulnerability in Windows Phone Internet Explorer

Check whether the flaw in CSS function in Internet Explorer allows attackers to **gain full access** over the phone through remote code execution.



BlackBerry Pen Testing

Follow the BlackBerry pen testing steps mentioned here to test your blackberry device to determine the potential vulnerabilities and to find the **security flaws** before an external attacker finds and exploits them:

Step 1: Perform blackjacking on the BlackBerry

Use **BBProxy** tool to hijack the BlackBerry connection.

Step 2: Check for flaws in the application code signing process

Obtain **code-signing** keys using prepaid credit cards and false details, sign a malicious application, and publish it on the BlackBerry app world.

Step 3: Perform an email exploit

Send an email or message to trick a user to download a malicious .cod application file on the BlackBerry device.

Step 4: Perform a DoS attack

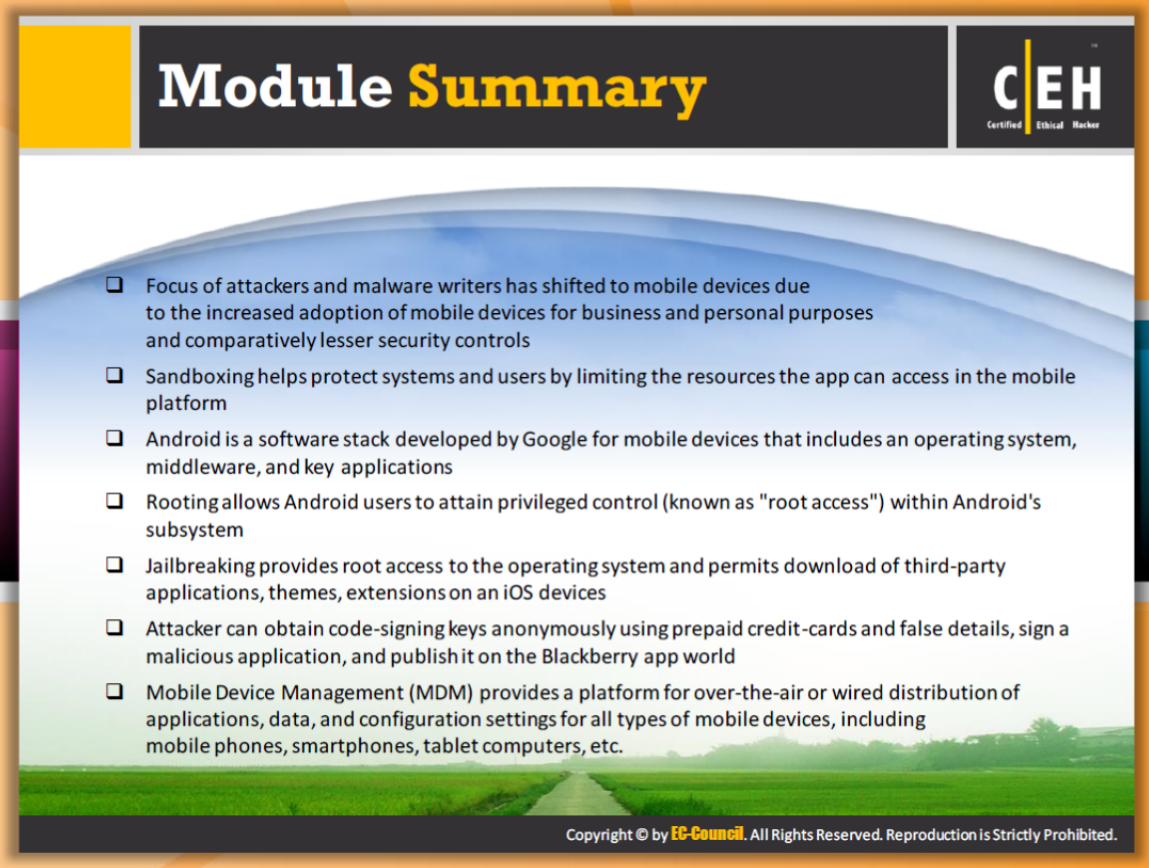
Try sending malformed **Server Routing Protocol (SRP)** packets from the BlackBerry network to the router to cause a **DoS attack**.

Step 5: Check for vulnerabilities in the BlackBerry Browser

Send maliciously crafted web links and trick users to open links containing malicious web pages on the BlackBerry device.

Step 6: Search for password protected files

Use tools such as Elcomsoft Phone Password Breaker that can recover password protected files and backups from BlackBerry devices.



Module Summary

- The focus of attackers and malware writers has shifted to mobile devices due to the increased adoption of mobile devices for business and personal purposes and comparatively lesser security controls.
- Sandboxing helps protect systems and users by limiting the resources the app can access in the mobile platform.
- Android is a software stack developed by Google for mobile devices that includes an operating system, middleware, and key applications.
- Rooting allows Android users to attain privileged control (known as "root access") within Android's subsystem.
- Jailbreaking provides root access to the operating system and permits download of third-party applications, themes, and extensions on iOS devices.
- Attacker can obtain code-signing keys anonymously using prepaid credit-cards and false details, sign a malicious application and publish it on the BlackBerry app world.

- Mobile Device Management (MDM) provides a platform for over-the-air or wired distribution of applications, data, and configuration settings for all types of mobile devices, including mobile phones, smartphones, tablet computers, and so on.