KaozhengPro

IT認證考試題庫

專業平臺

考證寶提供最新考古題與模擬試題
協助您高效通過認證考試

www.kaozhengpro.com

Exam : **CMPen-iOS**

Title : Certified Mobile Pentester (CMPen) – iOS

Version : DEMO

1.How can you inspect the sandboxed container of a third-party iOS app to identify sensitive data storage practices?
A. See the Explanation.
Answer: A
Explanation:
1. Jailbreak the iOS Device – Use checkra1n or unc0ver to jailbreak the device, granting file system access.
2. Connect via SSH – Retrieve the device's IP (Settings > Wi-Fi > ?) and connect:
ssh root@<device-ip>
3. Locate the App's Sandbox – List apps with: ls /var/mobile/Containers/Data/Application/
4. Identify the Target App – Use frida-ps -Uai to find the app's bundle ID and match it with directories.
5. Analyze Data Storage – Examine Documents/, Library/, and tmp/ for sensitive files using:
find . -type f -exec cat {} \;
This reveals whether the app stores credentials, PII, or encryption keys insecurely.

2.How can you extract and analyze an iOS app's Keychain data to check for security vulnerabilities?
A. See the Explanation.
Answer: A
Explanation:
1. Jailbreak the Device – Install Cydia and add the repository for Keychain-Dumper.
2. Install Keychain-Dumper – Use: apt-get install keychain-dumper
3. Dump Keychain Entries – Run: keychain_dumper
4. Analyze Output – Look for sensitive credentials stored under kSecAttrAccessibleAlways, which indicates insecure storage.
5. Mitigation – Developers should use kSecAttrAccessibleWhenUnlocked or stronger settings to prevent unauthorized access.

3.How can you test whether an iOS app correctly enforces permissions when accessing user data (e.g., Contacts, Location, Photos)?
A. See the Explanation.
Answer: A
Explanation:
1. Install the App on a Jailbroken iOS Device – Ensure root access.
2. Use Frida to Hook System Calls – Attach Frida to the target app: frida -U -n com.example.app -i
3. Monitor System Calls – Intercept permission requests:
Interceptor.attach(Module.findExportByName(null, "objc_msgSend"), { onEnter: function(args) {
console.log("Permission Request:", ObjC.Object(args[1]).toString());
}
});
4. Run the App and Test Functionality – Verify if permissions are correctly enforced before data access.
5. Mitigation – Ensure the app explicitly checks and requests permissions using Apple's Privacy Framework before accessing sensitive data.

4.How can you check if an iOS app uses improper ATS (App Transport Security) settings that could lead

to insecure communication?
A. See the Explanation.
Answer: A
Explanation:
1. Extract the App's Info.plist File – Use:
plutil -p /var/mobile/Containers/Data/Application/<APP-ID>/Library/Preferences/com.example.app.plist
2. Check ATS Settings – Look for:
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key>
<true/>
</dict>
This setting disables ATS, allowing insecure connections.
3. Intercept Traffic – Use Burp Suite or mitmproxy to confirm HTTP traffic: mitmproxy -p 8080
4. Mitigation – Developers should enforce HTTPS by setting NSAllowsArbitraryLoads to false.

5.How can you dynamically analyze an iOS application for security flaws using Frida?
A. See the Explanation.
Answer: A
Explanation:
1. Ensure Device is Jailbroken – Install Frida on the iOS device and on your computer.
2. List Installed Apps – Run:
frida-ps -U
3. Attach to the Target App – Use: frida -U -n com.example.app -i
4. Inject a Hook to Intercept API Calls – Example for logging sensitive data:
Interceptor.attach(Module.findExportByName(null, "objc_msgSend"), { onEnter: function(args) {
console.log("Intercepted:", ObjC.Object(args[1]).toString());
}
});
5. Monitor and Log Output – Analyze function calls that handle sensitive user data.