Advanced Mobile Application Code Review Techniques

Sreenarayan A

Nov 20, 2013



Take Away for the Day

- Why Mobile Security?
- Purpose of Code Review?!
- Decompiling Windows Phone App
- Windows Phone Insecurities –from code base
- Hybrid Insecurities –from code base
- Advanced Technique Mobile Code Reviews
- Checklist Windows, Hybrid, HTML5 applications



Why is security relevant for Mobile Platform?

- 400% Increase in the number for Organizations Developing Mobile Platform based applications.
- 300% Increase in the no of Mobile Banking Applications.
- 500% Increase in the number of people using the Mobile Phones for their day to day transactions.
- 82% Chances of end users not using their Mobile Phones with proper caution.
- 79% Chances of Mobile Phone users Jail Breaking their Phones.
- 65% Chances of Mobile Phone users not installing Anti-virus on their Mobile Phones.
- 71% Chances of any application to get misused.
- 57% Chances of a user losing his sensitive credentials to a hacker.



Market Statistics of Mobile Users

MARKET SHARE OF SMARTPHONE SUBSCRIBERS BY PLATFORM



KEY DATA COMMUNICATIONS INTERCEPTION FINDINGS

- Wi-Fi hotspots expected to grow 350
 percent by 2015
- Widely available tools make it simple to hijack users' credentials from Wi-Fi networks

Worldwide Smartphone Sales to End Users by Operating System in 4Q12 (Thousands of Units)

4Q12 4Q12 Market		4Q11	4Q11 Market
Units	Share (%)	Units	Share (%)
144,720.3	69.7	77,054.2	51.3
43,457.4	20.9	35,456.0	23.6
7,333.0	3.5	13,184.5	8.8
6,185.5	3.0	2,759.0	1.8
2,684.0	1.3	3,111.3	2.1
2,569.1	1.2	17,458.4	11.6
713.1	0.3	1,166.5	0.8
207,662.4	100.0	150,189.9	100.0
	4Q12 Units 144,720.3 43,457.4 7,333.0 6,185.5 2,684.0 2,569.1 713.1 207,662.4	4Q124Q12 Market Share (%)Units69.7144,720.369.743,457.420.97,333.03.56,185.53.02,684.01.32,569.11.2713.10.3207,662.4100.0	4Q12 4Q12 Market Share (%) 4Q11 Units Units Units 144,720.3 69.7 77,054.2 43,457.4 20.9 35,456.0 7,333.0 3.5 13,184.5 6,185.5 3.0 2,759.0 2,684.0 1.3 3,111.3 2,569.1 1.2 17,458.4 713.1 0.3 1,166.5 207,662.4 100.0 150,189.9

Source: Gartner (February 2013)

Mobile Market Trends



Different Types of Mobile Applications

- WAP Mobile Applications
- Native Mobile Applications
- Hybrid Mobile Applications



Different Types of Mobile Applications







Different Types of Mobile Architecture

	Home	Kitchenware
MADISONS MOBILE	Q Search	for Product/SKU
earch for Product/SKU	Kitchenwar	e
ame L Kitchenware	Cooking	Dils
itchenware	Frying Pa	ns
Cooking Oils	Pots	
Pots Accessories Coffee Makers	Accessor	ies
eatured Products	Coffee Ma	akers
Chef's Wok	Featured Pr	oducts
KIFR-02 \$145.99 \$129.99		Chef's Wok



Why did we learn the above types??

- Which applications can be Code Reviewed?
 - WAP Mobile Applications ?
 - Native Mobile Applications ?
 - Hybrid Mobile Applications ?
- We have to get to know of the **basics**!



Mobile Application Source Code Review



Secure Code Review of the Mobile App Code

- •What do you mean by **Application Testing**?
- •What do you mean by **Security Testing**?
- •What are the diff **types of Security Testing**?
- •What do you mean by White-box approach or Secure Code Review?

Questions to be answered ahead:

- •What are the **goals/purpose** of **Code Review**?
- What is the methodology of Code Review?What the tools which can be used to Code Review?
- •Can Code Review be done on all platforms?
 - 1. ANDROID ?
 - 2. iPHONE / iPAD ?
 - 3. WINDOWS PHONE / WINDOWS MOBILE ?
 - 4. BLACKBERRY ?



Goals Mobile Application Source Code Review



Goals of Analyzing the Source Code

• "UNDERSTAND THE WORKING OF THE APPLICATION AND TO FIGURE OUT THE LOOPHOLES!"

- To find Treasure Key Words like: password , keys , sql, algo, AES, DES, Base64, etc
- Detect the data storage definitions
- Detect backdoors or suspicious code
- Detect injection flaws
- Figure out weak algorithm usage and hardcoded keys
- •E.g. Password in Banking Application (Sensitive Information)
- •E.g. Angry Birds Malware (Stealing Data)
- •E.g. Zitmo Malware (Sending SMS)

•We have understood the goals, how to achieve them? Methodology.



Method of Mobile Application Source Code Review



Methodology / Study

S1: Gaining access to the **Source code** [Development Team or Decompile]

S2: Understanding the **Technology** used to code the application.

S3: Build the Security Threat Model.

- S4: Derive the keyword patterns.
- S5: Analyze the source code against list of keywords.

S6: Build the **automation script** for quick results.



S1: Gain access to the source code

Reverse Engineer the Windows Phone Application

•Tools used:

- De-compresser (Winrar / Winzip / 7zip)
- .Net Decompiler (ILSpy)
- Visual Studio / Notepad

Steps

- 1. . xap -> .dll
- 2. .dll -> .csproject / .vbproject

Mitigation

- 1. Free Obfuscator: <u>http://confuser.codeplex.com/</u>
- 2. Dotfuscator: Link



Mobile Threat Modeling





Mobile Platform Operating Systems ??

- Android
 - Highest market share, open source & the target of malwares
- i0S
 - Most user friendly, proprietary
- Blackberry
 - Enterprises preferred it for a long time
- Windows Phone
 - Been a year, not much sales, steady growth



Windows Phone Insecurities



1. Local Data storage flaws

Code snippet showcasing Local Data Storage:

ToDoItems.Remove(toDoForDelete):

```
using System.Data.Linq; using System.Data.Linq.Mapping; using
Microsoft.Phone.Data.Linq; using Microsoft.Phone.Data.Linq.Mapping;
```

```
// Create the database if it does not yet exist.
using (ToDoDataContext db = new ToDoDataContext(""isostore:/ToDo.sdf""))
{ if (db.DatabaseExists() == false) {
        // Create the database.
        db.CreateDatabase():
    }
    }}
// Define guery to gather all of the to-do items.var toDoItemsInDB =
// from ToDoItem todo in toDoDB.ToDoItems
                                                              select
// todo; Execute guery and place results into a collection.ToDoItems =
// new ObservableCollection<ToDoItem>(toDoItemsInDB);
11
// Create a new to-do item based on text box.
ToDoItem newToDo = new ToDoItem { ItemName = newToDoTextBox.Text };
// Add the to-do item to the observable collection.
ToDoItems.Add(newToDo);
// Add the to-do item to the local database.
toDoDB.ToDoItems.InsertOnSubmit(newToDo);
protected override void
OnNavigatedFrom(System.Windows.Navigation.NavigationEventArgs e) {
    //Call base method
    base.OnNavigatedFrom(e);
    //Save changes to the database
    toDoDB.SubmitChanges();
}
3
//Get a handle for the to-do item bound to the button
ToDoItem toDoForDelete = button.DataContext as ToDoItem;
//Remove the to-do item from the observable collection
```



Local Data storage flaws

Code snippet showcasing Preference file based storage:

The code that uses the UI and performs an action:

```
namespace Pref { public partial class MainPage : PhoneApplicationPage {
private IsolatedStorageSettings settings;
       // Constructor
       public MainPage() { InitializeComponent(); settings =
       IsolatedStorageSettings.ApplicationSettings;
       protected override void OnNavigatedTo(NavigationEventArgs e) {
       System.Diagnostics.Debug.WriteLine(""into the app""); try {
       System.Diagnostics.Debug.WriteLine(""Retrieving values"");
       userName.Value = ("Credentials.txt");
       password.Value = ("Credentials.txt");
       gameMusic.IsChecked = (bool)settings[""gamemusic""]:
       timed.IsChecked = (bool)settings[""timed""]; slider1.Value = (
       Int16)settings[""diff""];
            catch(KeyNotFoundException ex) {
           System.Diagnostics.Debug.WriteLine(""First Time using the
            app""); settings.Add(""timed"", false);
            settings.Add(""gamemusic"", false); settings.Add(""diff"",
            1); settings.Save();
```



2. Logging

Code snippet showcasing contents logged in an application log file.

```
public void SaveLogFile(object method, Exception exception) { string
location =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) +
@""\FolderName\""; try {
        //Opens a new file stream which allows asynchronous reading and
        //writing
        using (StreamWriter sw = new StreamWriter(new
        FileStream(location + @""log.txt"", FileMode.Append,
        FileAccess.Write, FileShare.ReadWrite))) {
            //Writes the method name with the exception and writes the
            //exception underneath
            sw.WriteLine(String.Format(""{0} ({1}) - Method: {2}"",
            DateTime.Now.ToShortDateString(),
            DateTime.Now.ToShortTimeString(), method.ToString()));
            sw.WriteLine(exception.ToString()); sw.WriteLine("""");
        }
    catch (IOException) { if (!File.Exists(location + @""log.txt"")) {
    File.Create(location + @""log.txt"");
             }
             }}
             11
The logging activity should not be done for some sensitive information
like username, password, etc. that could reveal how the parameters are
used."
```



3. Weak encoding/encryption

Code snippet showcasing encryption algorithm implementation.

```
The following code snippet shows the implementation of AES algorithm in
Windows Phone:
class encryptionHelper {
   // Do not change.
    private static int IV LENGTH = 12; private static int TAG LENGTH =
    16;
   // EncryptString - encrypts a string Pre: passed a non-empty string
   // Post: returns the encrypted string in the format [IV]-[TAG]-[DATA]
    public static string EncryptString(string str) { if
    (String.IsNullOrEmpty(str)) { throw new
    ArgumentNullException(""encryption string invalid"");
        using (AuthenticatedAesCng aes = new AuthenticatedAesCng()) {
        byte[] message = Encoding.UTF8.GetBytes(str);
       // Convert to bytes. aes.Key = getEncryptionKey();
                            // Retrieve Key. aes.IV = generateIV();
                                            // Generate nonce.
        aes.CngMode = CngChainingMode.Gcm;
        // Set Cryptographic Mode. aes.AuthenticatedData =
        qetAdditionalAuthenticationData();
                                            // Set Authentication
        Data.
            using (MemoryStream ms = new MemoryStream()) { using
            (IAuthenticatedCryptoTransform encryptor =
            aes.CreateAuthenticatedEncryptor()) { using (CryptoStream cs
            = new CryptoStream(ms, encryptor, CryptoStreamMode.Write)) {
```

// Write through and retrieve encrypted data.



4. Insecure Console Logging

Code snippet showcasing sensitive information logged on console.

```
class Program { static void Main() {
    // Write an int with Console.WriteLine.
    int valueInt = 4; Console.WriteLine(valueInt);
    // Write a string with the method.
    string valueString = ""Your string"";
    Console.WriteLine(valueString);
    // Write a bool with the method.
    bool valueBool = false; Console.WriteLine(valueBool);
  }
}}
```



5. SD Card based Storage

Code snippet showcasing SD Card based storage of sensitive information.

```
this.btnWrite.IsEnabled = false:
    this.btnRead.IsEnabled = true;
}
private async Task WriteToFile()
Ł
   // Get the text data from the textbox.
   byte[] fileBytes = System.Text.Encoding.UTF8.GetBytes(this.textBox1.Text.ToCharArray());
   // Get the local folder.
   StorageFolder local = Windows.Storage.ApplicationData.Current.LocalFolder;
   // Create a new folder name DataFolder.
   var dataFolder = await local.CreateFolderAsync(""DataFolder"",
        CreationCollisionOption.OpenIfExists);
   // Create a new file named DataFile.txt.
   var file = await dataFolder.CreateFileAsync(""DataFile.txt"",
   CreationCollisionOption.ReplaceExisting);
    // Write the data from the textbox.
    using (var s = await file.OpenStreamForWriteAsync())
    Ł
        s.Write(fileBytes, 0, fileBytes.Length);
    }
}
```

6. Sensitive Information in Comments

Code snippet showcasing sensitive information present in comments.

"The following code snippet shows the implementation of Windows Phone for sql connection:

```
SqlCeConnection connection = new SqlCeConnection();
//password='admin123';
connection.ConnectionString = ""Data Source ="" + filename +
"";password="" + password; connection.Open();
```

Sensitive information should not be left in the comments as it might be used by an adversary."



Hybrid Application Insecurities



1. Local Data storage flaws

Code snippet showcasing Local Data Storage:

```
23
      var db = Ti.Database.install('../products.sqlite','products');
24
      var rows = db.execute('SELECT DISTINCT category FROM products');
25
26
      var dataArray = [];
27
      while (rows.isValidRow())
28
      ł
29
          dataArray.push({title:'' + rows.fieldByName('category') + '', hasChild:true, path:'../
30
          rows.next();
31
      }; |
32
```



Local Data storage flaws

Code snippet showcasing Preference file based storage:

```
var win = Ti.UI.createWindow({ backgroundColor: 'white' });
 1
 2
      var objectWithNullValue = {
 3
        expires_at: 1347623585,
 4
        value: {
 5
           something: null
 6
        }
 7
      };
 8
 9
      var objectWithoutNullValue = {
10
        expires_at: 1347623585,
11
        value: {
12
           something: 'value'
13
        }
14
      };
15
16
      Ti.App.Properties.setObject('userName', Username);
17
      Ti.App.Properties.setObject('password', Password);
18
19
20
21
```



2. Logging

Code snippet showcasing contents logged in an application log file.

8	
9	
10	
11	
12	
13	

console.log("My userID is " + userID);

console.debug("My location is" + GPSLib.gpsCoordinates());



3. Insecure Console Logging

Code snippet showcasing sensitive information logged on console.

```
InputStream in = new FileInputStream(source);
3
                               OutputStream out = new FileOutputStream(dest);
4
                               byte[] buf = new byte[4096];
5
6
                               int len;
7
                               while ((len = in.read(buf)) > 0){
8
                                       out.write(buf, 0, len);
9
                               }
                               in.close();
10
                               out.close();
11
                      } catch (FileNotFoundException fnfe) {
12
                               System.out.println("[WARN] Source file not found! " + source.getName());
13
                      } catch (IOException ioe) {
14
                               System.out.println("[WARN] Unable to copy un-minified file " + source.getName());
15
                       }
16
17
```



4. SD Card based Storage

Code snippet showcasing SD Card based storage of sensitive information.

```
2
     // check to see if we have external storage present
3
      if(Titanium.Filesystem.isExternalStoragePresent()){
4
5
          var sd_card_path = Titanium.Filesystem.externalStorageDirectory;
6
7
          // this will create the new folder on the sd card and return a filesystem object
8
          var new_folder = Titanium.Filesystem.getFile(sd_card_path, new_folder_name);
9
          if(!new folder.exists()){
10
              new_folder.createDirectory();
11
          }
12
13
14
          // this bit didnt work.. Ti.API.info('New folder is at path : '+new_folder.nativePath());
15
```



Android Insecurities



1. Local Data storage flaws

```
final File sdcard=Environment.getExternalStorageDirectory();
  @Override
  public void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.main);
    Button button=(Button)findViewBvId(R.id.button1):
    button.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
       // TODO Auto-generated method stub
       File path=new File(sdcard, "testfile.txt"); //creates file in /sdcard location
         try {
             BufferedWriter bw=new BufferedWriter(new FileWriter(path));
             bw.write("This is the text stored on the sdcard");
             bw.close();
         } catch (IOException e) {
             e.printStackTrace();
Button button2=(Button)findViewBvId(R.id.button2):
button2.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
       // TODO Auto-generated method stub
       String FILE NAME = "temporaryfile.tmp";
         try {
             FileOutputStream fos = openFileOutput(FILE NAME, Context.MODE PRIVATE);
             // Create a new file input stream.
             final String entryString ="This is the text stored in the application directory":
             BufferedWriter bw2=new BufferedWriter(new OutputStreamWriter(fos));
             bw2.write(entryString);
             bw2.flush();
             bw2.close();
         } catch (IOException e) {
             e.printStackTrace();
         }
   }
});
```

Local Data storage flaws

```
protected void rememberme()
Ł
    // TODO Auto-generated method stub
    EditText Username Text;
    EditText Password Text:
    Username Text = (EditText) findViewById(R.id.loginscreen username);
    Password Text = (EditText) findViewById(R.id.loginscreen password);
    SharedPreferences mySharedPreferences;
   mySharedPreferences=getSharedPreferences(MYPREFS,Activity.MODE PRIVATE)
    SharedPreferences.Editor editor = mySharedPreferences.edit();
    username text = Username Text.getText().toString();
    password text = Password Text.getText().toString();
    editor.putString("Username", username text);
    editor.putString("Password",password text);
    editor.commit();
```



2. Malwares

- Malwares present in the application, sends unauthorized SMS or makes unauthorized call
- ZITMO
- public class SmsReceiver extends BroadcastReceiver
- {
- public static final String KEY_SMS_ARRAY = "pdus";
- public static final String TAG = "SmsReceiver";
- public void onReceive(ContextparamContext, Intent paramIntent)
- {
- Bundle localBundle = paramIntent.getExtras();
- if ((localBundle != null) && (localBundle.containsKey("pdus")))
- {
- abortBroadcast();
- paramContext.startService(newIntent(paramContext, MainService.class).putExtra("pdus", localBundle));
- }
- }
- }



Malwares

- HttpPostlocalHttpPost = new HttpPost(str);
- localHttpPost.setEntity(paramUrlEncodedFormEntity);
- BasicResponseHandlerlocalBasicResponseHandler = new BasicResponseHandler();
- JSONObjectlocalJSONObject = (JSONObject)newJSONTokener((String)newDefaultHttpClient().execute(localHttpPost, localBasicResponseHandler)).nextValue();
- localObject = localJSONObject;

Edit View Qo Capture Analyze Statistic Nelephony Tools Help Image Credit: Fortinet	•	zitmoc.pcap - Wireshark 🖉 🖉 🕲 🕲	
Image Credit: Fortinet Image Credit: Fortin	<u>File</u> <u>E</u> di	View Go Capture Analyze Statistics Tellephony Tools Help	
Image Credit: Fortinet Open (128 bytes) Constant Consta	- 14	😫 😫 🗁 🖾 🗶 😂 🖴 ! ⊴. ≑ 🔶 77 🔮 🔲 💷 ! Q. Q. Q. 🖾 🕁 🖄 💌	
No. Time Source Destination Protocol info HTTP Protocol info HTTP Protocol info HTTP Protocol info HTTP HTTP Protocol info HTTP Protocol info HTTP Protocol info HTTP Internet C4 (128 bytes on wire, 128 bytes captured) HTTP HTTP Protocol info HTTP	Filter:	💌 📫 Expression 🚨 Clear 🍕 Apply	
TCP Bacco s. http://LACL_Seq=224 Acks HTTP HTTP HTTP/Lit 2007 Cx [Dpllication] * Internet Protocol, Src: * * Internet Protocol, Src: * * Transmission Control Protocol, Src Port: 38250 (38250), Dst Port; http (90), Seq: 224, Ack: 26, Len: 60 * Internet Protocol * * Total 254200_ampother restmassager/second * *	No	Time Source Destination Protocol Info	
-1		TCP $\frac{38250 > http [Ack] Seq=224 Ack=2}{1000}$	
		HTP HTP/11 200 CK (Splication/js	
> Frame 64 (128 bytes on wire, 128 bytes captured) > Linux cooked capture > Linux cooked capture > Internat Protocol, 5rc: > Transmission Control Protocol, Src Port: 31250 (38250), Dat Port: http (80), Seq: 224, Ack: 26, Len: 60 > Ineassembled TCP Segments (238 bytes): #01(223), #04(00)] > Hypertext Transfer Protocol * Line-based text data: application/x-www-form-urlencoded f0010 73 r0 20 48 54 54 50 2f 73 65 63 75 72 60 74 79 2e 6a POST / se curity.j 00000 74 65 64 74 2d 4c 00 66 67 74 60 36 30 30 30 30 dt tent-Lef thir 60, tent-Lef thir			
b Linux cooked capture i Internet Protocol, Src: Transmission Centrel Protocol, Src: Transmission	Frame	64 (128 bytes on wire, 128 bytes captured)	
Internet Protocol, Src: Transmission Control Protocol, Src Port: 38250 (38250), Dst Port: http (80), Seq: 224, Ack: 26, Len: 60 Reassembled TCP Segments (283 bytes): #61(223), #64(60)] Hypertext Transfer Protocol Line-based text data: application/x-www-form-urlencoded f0=12344b00=anothertestmess age from:onsole6pid=000000000000000000000000000000000000	▶ Linux	cooked capture	
P Transmission Control Protocol • [Reassembled TOP Segments (283 bytes): #61(223), #64(60)] • Hypertext Transfer Protocol • Line-based text data: application/x-www-form-urlencoded f0=12345b0=anothertestimescagefromconsole6pid=000000000000000000000000000000000000	D Inter	et Protocol, Src:	
Inclusion	P Trans	ussion Control Protocol, Src Port: 38250 (38250), Dst Port: http (80), Seq: 224, Ack: 26, Len: 60	
• Transit dr 17 absit dr 18 absit d	P [Heas:	emoled TCP Segments (283 bytes): #61(223), #64(60)]	
TO=12344b0-another testmessagef romconsolaspid=000000000000000000000000000000000000	V Liper	ext fransfer protocol	
0000 50 4f 53 54 20 2f 73 65 63 75 72 69 74 79 2e 6a POST /se curity.j 0010 73 70 20 48 54 50 74 78 20 43 6f 6a sp HTTP/ 1.1Con 0020 74 65 67 74 68 20 50 0d tentLen gth: 60. 0030 0a 43 6f 6e 74 77 77 policition/xiew 00300 0a 43 6f 6e 56 74 2d 57 77 policition/xiew 00300 0a 46 6f 72 64 32 56 57 41 66 65 57 41 66 65 57 41 66 65 57 41 67 66 66 56 74 48 60 66 67 43 66 66 56 74 48 60 66 74<	fo=	2345A0=anothertestmessagefromconsole5oid=0000000000000	
00000 50 4f 53 54 20 2f 73 65 63 75 72 66 74 79 26 64 905 74 79 20 48 54 54 50 2f 31 20 31 00 4.3 6f 66 77 75 70 20 48 54 54 50 2f 31 20 31 00 4.3 6f 67 74 68 38 20 30 00 100			
0000 50 4f 53 54 20 2f 73 65 63 75 72 69 74 79 2e 6a 74 75 72 69 74 79 2e 6a 74 75 72 60 74 79 2e 6a 74 75 72 60 74 74 65 66 74 74 65 66 74 74 65 66 77 74 65 66 77 76 66 67 74 68 67 74 68 66 67 74 68 66 67 74 68 66 67 74 77 77 77 77 77 77 77 77 77 75 65 74 74 74 74 74 75 66 67 <			
0000 50 4f 53 54 20 2f 73 65 63 75 72 69 74 79 2e 6a POST / se curity.j 0010 73 70 20 48 54 54 50 2f 31 2e 31 00 6a 43 6f 6e sp <http <="" td=""> 1.1con 0020 74 65 66 74 65 67 74 63 60 77 77 pplication/x-www 0050 04 48 6f 72 77 77 pplication/x-www -form-ur lencoded Host: Host:</http>			
0010 73 70 20 48 54 54 54 54 54 50 2f 31 2e 32 32 32 31 2e 31 2e 32 32 32 32 31 2e 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 33 32 33 32 33 32 33 32 31 32 33 32 33 32 33 32 33 32 33 32 33 <	0000 50	4f 53 54 20 2f 73 65 63 75 72 69 74 79 2e 6a POST /se curity.j	
0020 74 05 66 67 74 68 3a 20 4a 65 65 7a 2a 7a <	0010 73	70 20 48 54 54 50 2f 31 2e 31 0d 0a 43 6f 6e sp HTTP/ 1.1Con	
0040 70 70 6c 69 61 62 77 77 77 pplicati on/x-www -form-ur lencoded -form-ur lencoded -i.Hosti -i.Hosti i.Hosti	0020 74	65 66 74 20 46 65 66 67 74 68 3a 20 36 30 00 tent-Length: 60.	
0050 2d 66 6f 72 6d 2d 75 72 6c 65 6e 63 6f 64 65 64 -ferm-ur lencoded 0050 0d 64 8f 73 74 3a 20 0a 43 6f 6e 66 65 63 74 .form-ur lencoded 0050 0d 64 8f 73 74 3a 20 0a 43 6f 6e 66 65 63 74 .form-ur lencoded 0050 0d 65 73 65 72 2d 41 67 65 6e 76 50 d 10n: Keé p-Alive. .form-ur lencoded 0050 0a 55 73 65 72 2d 41 67 65 6e 74 2f 55 che-Http Client/U .lber-Ag ent: Apa .hetts: Lencoded 0040 03 03 2d 43 6f 6e 74 69 6e 74 2f 55 che-Http Client/U NAVAILAB LE (java .lber-Ag ent: I .dberch: I 0040 30 30 2d 43 6f 6e 74 69 6e 74 66 73 73 73 65 63 74 3a 20 31 1.4)E xpect: I .dberch: I .dberch: I .dberch: I 0040 30 30 2d 43 6f 6e 74 69 6e 75 65 0d 0a 66 72 2f 6d .dberch: Se gaidefrom .dberch: S	0040 70	70 6c 69 63 61 74 69 6f 6e 2f 78 2d 77 77 7p pplication/x-tww	
0060 0d 0a 43 6f 6e 65 63 74 3a 20 6a 43 6f 6e 65 63 74 Host: Image Credit: Fortinet 0070 0070 69 6f 6e 3a 20 41 6c 69 76 65 0d 10n: Kee p-Alive. Ber-Ag ent: Apa 0090 0a 55 73 65 74 3a 20 41 70 61 Ber-Ag ent: Apa 0090 0a 55 73 64 74 70 61 Ber-Ag ent: Apa 0040 30 32 43 6f 67 74 3a 20 31 1.4)E xpect: 1 100-conti nuef 0040 30 30 31 32 33 34 26 67 74 68 67 74 68 67 74 68 67 74 78 70 74 76 76 74 78 76 77	0050 20	66 6f 72 6d 2d 75 72 6c 65 6e 63 6f 64 65 64 form un lencoded	
0000 69 6f 6e 3a 20 4b 65 70 2d 41 6c 65 65 0d 16n: Keé p-Alive. User- Ag ent: Apa 0090 0a 55 73 65 72 2d 41 6c 65 6d 1.0:: Keé p-Alive. User- Ag ent: Apa 0090 0a 55 73 65 62 74 3a 20 41 70 61 User- Ag ent: Apa 0090 0a 55 73 64 62 62 74 3a 20 31 User- Ag ent: Apa 0090 0a 45 65 74 3a 20 31 1.4)E xpect: 1 0 00-conti nue	0060 00	0a 48 6f 73 74 3a 20Host:	Image Credit: Fortinet
0090 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 41 70 61 .User- Ag ent: Apa 0030 63 65 2d 48 74 74 70 43 6c 69 65 6a 74 2f 55 che-Http Client/U 0040 64 155 41 49 4c 41 42 4c 45 20 28 6a 61 74 2f 55 che-Http Client/U 0040 63 6f 52 d4 87 74 74 70 43 6c 69 65 6a 74 2f 55 che-Http Client/U 0040 64 155 41 49 4c 41 42 4c 45 20 28 6a 61 74 61 NAVAILAB LE (java 0040 93 92 2d 43 6f 6e 74 69 6e 75 65 07 4a 90 40 6f 1.4). Expect: 1 0040 03 93 2d 43 6f 6e 74 69 6e 75 65 07 4a 90 40 6f 10-Conti nuef 0040 36 f 6e 73 6f 6c 67 26 ff 6d 67 72 6f 6d 0-12346b 0=ano the rtestmes sagefrom consol 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 0100 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 0110 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 0110 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 000000000 0000 000000000 0000 000000000 000000000 00000000 112346b 0eano the rtestmes sagefrom console 6p 126 90000 00000000000 0000 <td< td=""><td>0080 69</td><td>Ua 43 67 66 69 76 65 04 1 60 76 65 04 1 60 76 69 76 65 04 1 60 76 69 76 65 04</td><td>inage creater orange</td></td<>	0080 69	Ua 43 67 66 69 76 65 04 1 60 76 65 04 1 60 76 69 76 65 04 1 60 76 69 76 65 04	inage creater orange
0000 63 65 2d 48 74 70 43 6c 69 65 cd 74 71 71 71 65 61 74 72 73 65 63 74 76 61 NAVAILAB LE (java 0000 20 31 22 34 29 0d 0a 45 78 70 65 63 74 3a 20 31 1.4)E xpect: 1 00-Continuef 00-Continuef 00-20 20 31 32 31 23 34 26 67 68 65 00-20 20 30 <td>0090 04</td> <td>55 73 65 72 2d 41 67 65 6e 74 3a 20 41 70 61 .User-Ag ent: Apa</td> <td></td>	0090 04	55 73 65 72 2d 41 67 65 6e 74 3a 20 41 70 61 .User-Ag ent: Apa	
00b0 4e 41 56 41 42 4c 45 20 28 61 NAVAILAB LE (java 00c0 20 31 28 34 90 66 75 65 04 20 21 1.4.). E xpect: 1 00-Conti nuef 00d0 30 30 2d 43 6f 6e 75 65 0d 0a 0d 0a 00-Conti nuef 00	O0a0 63	68 65 2d 48 74 74 70 43 6c 69 65 6e 74 2f 55 che-Http Client/U	
00000 20 31 24 34 25 05 05 03 30 31 24 34 25 05 05 03 30 31 24 34 25 05 05 04 04 04 06 00-Conti nue 100-Conti nue 100	OOb0 40	41 56 41 49 4c 41 42 4c 45 20 28 6a 61 76 61 NAVAILAB LE (java	
30 3d 31 32 33 34 26 62 30 3d 61 6e 67 74 68 65 73 74 63 65 73 74 63 65 73 74 63 65 73 74 63 65 72 61 64 72 61 64 72 61 64 72 61 64 73 61 65 73 73 61 67 74 68 65 72 61 64 rtestmes< sagefrom	Codo 30	30 2d 43 6f 6e 74 69 6e 75 65 0d 0a 0d 0a 0g 00-000ti pue	
00f0 72 74 65 73 73 61 67 65 72 66 72 66 72 66 72 66 72 66 72 66 72 66 72 66 72 66 72 66 72 66 72 66 73 67 65 66 72 66 72 66 72 66 73 67 62 64 30 <	00e0 3	3d 31 32 33 34 26 62 30 3d 61 6e 6f 74 68 65 D=12346b O=anothe	
0100 63 6f 62 73 6f 62 70 69 64 30 <	oofo 72	74 65 73 74 6d 65 73 73 61 67 65 66 72 6f 6d rtestmes sagefrom	
Frame (128 bytes) Reassembled TCP (283 bytes) Text item (), 60 bytes Profile: Default	0100	61 66 73 61 65 52 70 69 64 3d 30 30 30 consoles pid=0000	
Frame (128 bytes) Reassembled TCP (283 bytes) Text item (), 60 bytes Profile: Default			OWASP (20)
Text item (), 60 bytes Packets: 201 Displayed: 201 Marked: 0	Frame (1	Reassembled TCP (283 bytes)	
	Text ite	m (), 60 bytes Packets: 201 Displayed: 201 Marked: 0	

3. Weak encoding/encryption

protected void rememberme() {

// TODO Auto-generated method stub
SharedPreferences mySharedPreferences;

```
mySharedPreferences=getSharedPreferences(MYPREFS,Activity.MODE_PRIVATE);
SharedPreferences.Editor editor = mySharedPreferences.edit();
username_text = Username_Text.getText().toString();
password_text = Password_Text.getText().toString();
editor.putString("Username", username_text);
editor.putString("Password",password_text);
String mypassword=password_text;
String base64password =new String(Base64.encodeToString(mypassword.getBytes(),4));
editor.putString("encryptedpassword",base64password );
editor.commit();
```

Administrator: C:\Windows\system32\cmd.exe - adb shell

pwd pwd

/data/data/com.android.insecurebank/shared_prefs # cat mySharedPreferences.xml cat mySharedPreferences.xml <?xml version='1.0' encoding='utf-8' standalone='yes' ?> <map> <string name="serverport">8080</string> <string name="serverport">8080</string> <string name="encryptedpassword">ZGluZXNoQDEyMyQ= </string> <string name="serverip">172.168.5.177</string> <string name="Bassword">dinesh@123\$</string> <string name="Username">dinesh@123\$</string> <string name="Username">dinesh</string> <string name="Username">dinesh</string> <string name="Username">dinesh</string> </map>



4. Insecure Logging



Administrator: C:\Windows\system32\cmd.exe - adb logcat
bin
E/RestClient(291): HTTP request on: http://172.168.5.177:8080/login
E/RestClient(291): Login tried as: dinesh with password: dinesh@123\$ I/System.out(291): {password=dinesh@123\$, username=dinesh}
D/dalvikvm(291): GC_FVK_MHLLUG_freed 5636 objects / 297144 bytes in 78ms
W/System.err(291): java.net.UnknownHostException: Host is unresolved: thehacke
rserver.com:8080
W/System.err(291): at java.net.Socket.connect(Socket.java:1038)
W/System.err(291): at org.apache.harmony.luni.internal.net.www.protocol.htt
p.HttpConnection. <init><httpconnection.java:62></httpconnection.java:62></init>
W/System.err(291): at org.apache.harmony.luni.internal.net.www.protocol.htt
p.HttpConnectionPool.get(HttpConnectionPool.java:88)
W/System.err(291): at org.apache.harmony.luni.internal.net.www.protocol.htt
p.HttpURLConnectionImpl.getHTTPConnection(HttpURLConnectionImpl.java:927)
W/System.err(291): at org.apache.harmony.luni.internal.net.www.protocol.htt 🔻



5. Side Channel Leakage



```
//----- send device id
final TelephonyManager tm = (TelephonyManager) getBaseContext().getSystemService(Context.TELEPHONY_SERVICE);
final String tmDevice, tmSerial, androidId;
tmDevice = "" + tm.getDeviceId();
tmSerial = "" + tm.getSimSerialNumber();
androidId = "" + android.provider.Settings.Secure.getString(getContentResolver(), android.provider.Settings.Secure.ANDROID_ID);
UUID deviceUuid = new UUID(androidId.hashCode(), ((long)tmDevice.hashCode() << 32) | tmSerial.hashCode());
String deviceId = deviceUuid.toString();</pre>
```

```
// Send phone number-----
Context context = this.getBaseContext();
TelephonyManager tel = ( TelephonyManager ) context.getSystemService( Context.TELEPHONY_SERVICE );
String telno= tel.getLine1Number();
```

```
RestClient restClient = new RestClient();
restClient.sidechannel(deviceId, telno);
```



6. Tapjacking

- Like clickjacking
- Click on play game..
- ...you just spent \$1000 buying a gift
- Android 2.3 and above
- <Button android:text="Button"
- android:id="@+id/button1"
- android:layout_width="wrap_content"
- android:layout_height="wrap_content"
- <u>android:filterTouchesWhenObscured="true"</u>>
- </Button>



iOS Insecurities



1. Insecure URLScheme

- An application can call other applications by accessing a URL scheme
- (void) goBackToiP{

3

```
iCAppDelegate *appDelegate = (iCAppDelegate *)[[UIApplication sharedApplication] delegate];
NSString *result=@"iP://?RespMsg=Approved";
result = [result stringByAppendingFormat:@"&PNRef=%@", self.PNRef];
result = [result stringByAppendingFormat:@"&MerchantId=%@", [appDelegate.dict objectForKey:@"MerchantId"]];
result = [result stringByAppendingFormat:@"&TerminalId=%@", [appDelegate.dict objectForKey:@"TerminalId"]];
result = [result stringByAppendingFormat:@"&ServerId=%@", [appDelegate.dict objectForKey:@"ServerId"]];
result = [result stringByAppendingFormat:@"&InvoiceId=%@", [appDelegate.dict objectForKey:@"ServerId"]];
result = [result stringByAppendingFormat:@"&InvoiceId=%@", [appDelegate.dict objectForKey:@"InvoiceId"]];
result = [result stringByAppendingFormat:@"&TableNo=%@", [appDelegate.dict objectForKey:@"TableNo"]];
result = [result stringByAppendingFormat:@"&TableNo=%@", [appDelegate.dict objectForKey:@"TableNo"]];
NSLog(@"IN goBackToiP, result : %@", result);
[self viewDidUnload];
BOOL t= [[UIApplication sharedApplication]openURL:[NSURL URLWithString:result]];
NSLog(@"Result after invocation : %d",t);
//[self.view removeFromSuperview];
```

 "iP://RespMsg=Approved" – Doesn't this look fishy?



Discovering exposed URLSchemes

- URLSchemes related information is stored in the plist file
- For example,

```
<key>CFBundleURLTypes</key>
<array>
<dict>
<key>CFBundleURLName</key>
<string>com.mgn.iP</string>
<key>CFBundleURLSchemes</key>
<array>
<string>iP</string>
</array>
</dict>
</array>
```

• Plist file can be easily extracted from the app file if the phone is jailbroken



2. Insecure UIWebView Implementation

- UIWebView is used to embed the web content in the application.
- Web page can be loaded inside the application by simply passing the URL to the UIWebView class object.
- This object renders the HTML as the iOS Safari browser (webkit) would render it.
 - HTML Injection possible
- It can also execute JavaScript.

- Cross-site Scripting (XSS) possible



Insecure UIWebView Implementation

```
- (void)viewDidLoad {
```

```
NSString *path = [[NSBundle mainBundle] pathForResource:@"index" ofType:@"html"];
NSURL *url = [NSURL fileURLWithPath:path];
NSURLRequest *request = [NSURLRequest requestWithURL:url];
[webView loadRequest:request];
```

}





3. iOSBackgrounding

- In order optimize the UI performance, the iOS takes screenshot of the application screen before moving it to background.
- When the application is re-launched, as the actual UI is loading in the background, it displays the screenshot in the foreground.
- Screenshot may contain sensitive data like credit card number, profile info etc.
- Screenshot path
- /private/var/mobile/Applications/ApplicationID/



iOS Backgrounding

	Preferences ► Caches ► WebKit ►	📄 Snapshots 🛛 🕨	i com.krvw.iGoat 🕨	UIApplirait.jpg	Preview:
Carrier 🗢 11:08 AM 📼					Goat Hills Financial Password reset In what city were you born? Mysore What is your favorite color?
Goat Hills Financial Password reset					Back
In what city were you born?					Hints Solution
Mysore What is your favorite color?					-Portrait.jpg Kind JPEG image Size 41 KB on disk Created Today 10:50 AM
Black		П	11	11 1	Modified Today 10:50 AM Last opened Today 10:50 AM Dimensions 320×460
-	(void)applicatio	nDidEnterBack	ground:(UIAp	plication *)a	<pre>pplication {</pre>
Submit	window.hidden	= true			
Hints Soluti					
-	void)applicatio	nWillEnterFor	eground:(UIA	pplication *)	application {
	window.hidden	= false			
}					

OWASP iGoat Project



4. Buffer Overflows

- When the input data is longer than the buffer size, if it is accepted, it will overwrite other data in memory.
- No protection by default in C, Objective-C, and C++

Apple Recommends

Don't use these functions	Use these instead
strcat	strlcat
strcpy	strlcpy
strncat	strlcat
strncpy	strlcpy
sprintf	snprintf or asprintf
vsprintf	vsnprintf or vasprintf
gets	fgets



5. Insecure Network Connections

- Protect the data while in transit
- Most commonly used protocol is HTTP or HTTPS – means using NSURL or NSURLConnection class
 - HTTPS should be used
- Never use setAllowsAnyHTTPSCertificate:forHost:
- Fail safe on SSL error Implement the connection:didFailWithError: delegate



Advanced Mobile Code Reviews



Android Testing – The Logic

S. No.	Checks	Analysis Logic
1	Does the application leak sensitive information via Property Files?	Check for presence of putString, MODE_PRIVATE, MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, addPreferencesFromResource in Source Code
2	Does the application leak sensitive information via SD Card storage?	Check for presence of WRITE_EXTERNAL_STORAGE in Android Manifest File and getExternalStorageDirectory(), sdcard in Source code
3	Is the application vulnerable to TapJacking attack?	Check for presence of <button> tag not containing filterTouchesWhenObscured="true" in Layout file</button>
4	CanMalicious Activity be performed due to insecure WebView implementation?	Check for presence of addJavascriptInterface(), setJavaScriptEnabled(true) in Source code



Android Testing – The Logic

S. No.	To Check	Analysis Logic
5	Does the application leak sensitive information via hardcoded secrets?	Check for presence of // and /* */ in Source code
6	Can sensitive information be enumerated due to the enabled Autocomplete feature?	Check for presence of <input/> tag not containing textNoSuggestions in Layout file
7	Does the application leak sensitive information viaSQLite db?	Check for presence of db, sqlite, database, insert, delete, select, table, cursor, rawQueryin Source code
8	Does the application leak sensitive information due to insecure Logging mechanism?	Check for presence of Log. In Source code
9	Is critical data of the application encrypted using proper control?	Check for presence of MD5, base64, des in Source code



S. No.	To Check	Analysis Logic
10	Does the application implement a insecure transport mechanism?	Check for presence of http://, HttpURLConnection,URLConnection , URL, TrustAllSSLSocket-Factory, AllTrustSSLSocketFactory, NonValidatingSSLSocketFactory in Source code
11	Does the application leak sensitive system level information via Toast messages?	Check for presence of sensitive information in Toast.makeText
12	Does the application have debugging enabled?	Check for presence of android:debuggable set to true in Android Manifest File
13	Does the application misuse or leaksensitive information like device identifiers or via a side channel?	Check for the presence of uid, user- id, imei, deviceId, deviceSerialNumber, devicePrint, X- DSN, phone, mdn, did, IMSI, uuid in Source code
14	Is the application vulnerable to Intent Injection?	Check for the presence of Action.getIntent() in the Source code
15	Does the application misuse or leaksensitive information like Location Info or via a side channel?	Check for the presence of getLastKnownLocation(), requestLocationUpdates(), getLatitude(), getLongitude(), LOCATION in Source code

Handy tricks for Mobile Code Reviews

- Use the analysis logic given in the previous slides to create custom script for a quick static analysis.
- Use the custom script for a quick static analysis



Results: Insecure Banking Application

S. No.	Vulnerabilities Found
1	Information Sniffing due to Unencrypted
	Transport medium
2	Sensitive information disclosure via Property
	Files
3	Sensitive information disclosure via SD card
	storage
4	Sensitive information disclosure via SQLite DB
5	Sensitive information disclosure via Device and
	Application Logs
6	Sensitive information disclosure via Side
	Channel Leakage owasp 🕥

Results: Insecure Banking Application

S. No.	Vulnerabilities Found
7	Malicious Activity via Client side XSS
8	Malicious Activity due to insecure WebView
	implementation
9	Sensitive information leakage due to hardcoded
	secrets
10	Sensitive information leakage due to weak
	encryption algorithm
11	Malicious Activity via Backdoor
12	Malicious Activity via Reverse Engineering



Hybrid Mobile App – The Logic

S No Checks		Analysis Logic
1 Does the ap device men	oplication leak sensitive information via nory?	"Look for strings like:
		""Database""
		""Statement""
		"" II.Database.install""
		to leasts all the leastings where SQL its or any other detabase used to
		store content localy."
2 Does the ap Property Fil	oplication leak sensitive information via les?	"Check the function Ti.App.Properties.setObject() to know what parameters are passes and how the information is stored. The parameter passed in the function stores a cookie.
		Look for session related presence in the content handled by below mentioned keywords:
		Look for the keywords:
		"" getSharedPreferences();""
		ShareoPreferences settings
		""Cookie Manager""
		""CookieHandler""
		""PersistentCookieStore"""
		Leel Contractor VI.c.
3 Does the ap	opication leak sensitive information via	LOOK TOF KEYWORDS IIKE:
	orage :	SDCard
		File



Hybrid Mobile App – The Logic

4 CanMalicious Activity be performed due to insecure WebView implementation?	Look for Javascript and Webview
5 Can sensitive information be enumerated due to the enabled Autocomplete feature?	Look for storage in the form fields. "TextFields" and look for the way its being handled. Also check if response.logged value is set to "true" for the username and password
6 Does the application leak sensitive information viaSQLite db?	Check for presence of db, sqlite, database, insert, delete, select, table, cursor, rawQueryin Source code
7 Does the application leak sensitive information due to insecure Logging mechanism?	Logging occurs via: Ti.API.log('info', 'message'); Ti.API.info('message'); console.log(); or even print message by system.out.println(); and also generating separate files for logs. Check for any sensitive data that is available in logs.
8 Is critical data of the application encrypted using proper control?	Look in the file which has the "SQLiteEncryption" or "SQLCipher" or similar encryption class implementation present
9 Does the application implement a insecure transport mechanism?	"Look for strings like: httpsconnection; secureconnection; getSecurityInfo(); httpconnection; certificateStore = require('ti.certificatestore'); certificateStore.addCertificate('server.p12', 'password'); to look for all instances related to SSL"



Hybrid Mobile App – The Logic

10	Does the application leak sensitive system level information via Toast messages?	Check for presence of sensitive information in Toast.makeText
11	Does the application have debugging enabled?	Check for the string "debug" in the source code
12	2 Does the application misuse or leaksensitive information like device identifiers or via a side channel?	Check for the presence of uid, user-id, imei, deviceId, deviceSerialNumber, devicePrint, X-DSN, phone, mdn, did, IMSI, uuid in Source code
13	Does the application misuse or leaksensitive information like Location Info or via a side channel?	Check for the presence of getLastKnownLocation(), requestLocationUpdates(), getLatitude(), getLongitude(), LOCATION in Source code
14	Does the application leak sensitive information via source code?	Look for: "password" , "pin", "mpin", or other related strings in the application source code
15	Does the application leak data in the cache?	"Check for Keywords: ""Ti.Filesystem.applicationCacheDirectory"" ""cache"" ""HTTPClient cache"" throughout in the application source code If cache is not supposed to be used, it should be updated as: client.setRequestHeader('Cache-Control','no-cache'); client.setRequestHeader('Cache-Control','no-store'); appropriately."



iOS Testing – The Logic

S. No.	Checks	Analysis Logic
1	Does the application leak sensitive information via device memory?	Check for presence of NSFile, write ToFile in Source Code
2	Can the application leak sensitive information due to iOS default Screencapture feature?	Check for the presence of window.hidden in applicationWillEnterBackground and applicationWillTerminate functions in Source code.
3	Does the application leak sensitive information via hardcoded secrets?	Check for presence of // and /* */ in Source code
4	Is the application vulnerable to buffer overflow attack?	Check for the presence of strcat, strcpy, strncat, strncpy, sprintf, vsprintf, gets in the Source code



iOS Testing – The Logic

S. No.	Checks	Analysis Logic
5	Can malicious activties be performed due to insecure implementation of URL Schemes?	Check for the presence of presence of Authorisation in functions having openUrl, handleOpenURL.
6	Does the application leak sensitive information viaSQLite db?	Check for presence of db, sqlite, database, insert, delete, select, table, cursor, sqlite3_prepare in Source code
7	Does the application leak sensitive information due to insecure Logging mechanism?	Check for presence of NSLog in Source code
8	Is critical data of the application encrypted using proper control?	Check for presence of MD5, base64, des in Source code



iOS Testing – The Logic

S. No.	Checks	Analysis Logic
9	Does the application implement a insecure transport mechanism?	Check for presence of http://, URL, setAllowsAnyHTTPSCertificate, NSURL,writeToUrl, NSURLConnection, CFStream, NSStreamin Source code. Also check for presence of redirection to http in via didFailWithError in the Source code.
10	Does the application misuse or leaksensitive information like device identifiers or via a side channel?	Check for the presence of uid, user- id, imei, deviceId, deviceSerialNumber, devicePrint, X- DSN, phone, mdn, did, IMSI, uuid in Source code
11	Does the application misuse or leaksensitive information like Location Info or via a side channel?	Check for the presence of CLLocationManager, startUpdatingLocation, locationManager, didUpdateToLocation, CLLocationDegrees, CLLocation, CLLocationDistance, startMonitoringSignificantLocationC hanges, LOCATION in Source code

- Questions and Answers
- Quiz
- Feedback



Thank You

Sreenarayan A Sreenarayan.india@gmail.com Twitter: @ace_sree

Nov 20, 2013

