OS-level Side Channels without Procfs: Exploring Cross-App Information Leakage on iOS

Xiaokuan Zhang¹, Xueqiang Wang², Xiaolong Bai³, Yinqian Zhang¹ and XiaoFeng Wang²

¹The Ohio State University, ²Indiana University Bloomington, ³Tsinghua University
Mobile Side-Channel Attacks

- Side-channel Attack: make use of seemingly harmless information to infer sensitive information
OS-level Side-Channel Attacks on Android

- Malicious app running in the background, calling APIs
  - Procfs: system statistics
    - virtual/physical memory, network traffic, CPU usage info, ...

```bash
zlk@zlk-VirtualBox:$ ls /proc
1  1498  1776  1957  2655  2226  2421  4  526  65  769  96
10  15   18  1961  2661  2230  2476  401  53  66   77  97
1056  150  1870  1962  2664  2245  2499  47  54  67  78  acpi
11  154  1881  1966  2690  2246  25  471  55  68  8  asound
1102  1542  1886  1967  2699  2251  2524  475  555  69  82  buddyinfo
1134  155  19  1980  21  2255  2535  48  56  693  866  bus
1197  156  1911  1984  2129  2271  2544  49  561  7  870  cgroups
1217  157  1912  2  2143  2277  2545  493  57  70  877  cmdline
1221  158  1913  20  2164  23  2558  5  58  71  878  consoles
1234  16  1916  2041  2176  2364  26  50  59  714  881  cpupinfo
1286  1655  1921  2045  2189  2373  28  503  6  72  9  crypto
13  169  1925  2046  2198  2387  29  507  60  726  938  devices
1308  17  1929  2047  22  2399  3  51  61  73  945  diskstats
1333  170  1931  2048  22  2400  24  30  517  62  74  95  dna
14  1704  1941  2051  2205  2404  31  52  63  75  951  driver
148  1774  1954  2054  2207  2411  397  525  64  76  956  execdomains
```
OS-level Side-Channel Attacks on iOS

- No Procfs providing system stat
- No unauthorized cross-app query

Is it possible to conduct OS-level side-channel attacks on iOS?
Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion
Threat Model

• Monitoring app:
  • User downloads it from App Store
  • Audio player
New Attack Vectors

• Host_statistics64(): Global usage of memory resources

```c
kern_return_t host_statistics64(host_t host_priv, host_flavor_t flavor,
host_info64_t host_info64_out,
mach_msg_type_number_t *host_info64_outCnt);
```

• Getifaddrs():

```c
int getifaddrs(struct ifaddrs **ifap);
```

• [NSFileManager fileExistsAtPath:]: The existence of a file/directory

```c
- (BOOL)fileExistsAtPath:(NSString *)path;
```
Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion
Classifying User Activities --- Example Trace

- Calling APIs to get time series A
  - Host_statistics64()
  - Getifaddrs()
- Plotting diff series: A[i] – A[i-1]

Time series leak information!!!
Classifying User Activities --- Example Trace

How to combine multiple time series to perform inference attacks?
Classifying User Activities --- Example Trace

How to combine multiple time series to perform inference attacks?

• Requirements:
  • Combining multiple time series
  • Reducing the dimension

• Major components:
  • SAX (Keogh et al., 2002)
  • BOP (Lin et al., 2009)
  • LibSVM (Chang et al., 2011)
Classifying User Activities --- Case Studies

• Device: jailbroken iPhone 7 with iOS 10.1.1

• Automated using Cycrypt

• Monitoring app:
  • running in the background
  • calling APIs at a rate of 1000/s
Classifying User Activities --- Case Studies

• Foreground Apps:
  • 100 apps from Top Charts + 20 pre-installed apps
  • Top N accuracy: the percentage of the test samples being correctly labeled by one of the top N predicted classes by the classifier

- Top 1: 89.2%
- Top 2: 97.5%
Classifying User Activities --- Case Studies

- Safari Websites

84.5%
Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion
Detecting Sensitive In-App Activities

Blockchain.info
Detecting Sensitive In-App Activities --- Attack Methods

• Identify critical events

• Correlates with public records
Detecting Sensitive In-App Activities --- Case Studies

• Target: *Blockchain Wallet* App

• Goal: identify *payment* event (idx: 0)

\[
d(\vec{X}_t, \vec{S}_t) = \sum_{k=1}^{l} \frac{1}{w_k} \cdot DTW(\vec{X}_t^k, \vec{S}_t^k)
\]
Detecting Sensitive In-App Activities --- Case Studies

- **Target**: Blockchain Wallet App
- **Goal**: identify *payment* event (idx: 0)
- Normalize the distance per row using cell(i,i) as the base (diagonal)

\[ d(\vec{X}_t, \vec{S}_t) = \sum_{k=1}^{l} \frac{1}{w_k} \cdot \text{DTW}(\vec{X}_t^k, \vec{S}_t^k) \]
Detecting Sensitive In-App Activities --- Case Studies
Detecting Sensitive In-App Activities --- Case Studies

**A** sent 0.0035 BTC to B (1EwB...), The rest went to C (1Fbr...)

**C** sent 0.001 BTC to E (1yNT...), The rest went to D (1ANE...)

**D** sent 0.0028 BTC to F (1CeN...), The rest went to G (16rU...)

Detecting Sensitive In-App Activities --- Case Studies

• Other Targets: Venmo / Twitter
Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion
Bypassing Sandbox Restrictions --- Attack Methods

• Device: non-jailbroken iPhone 7 with iOS 10.2.1

• Execution time of FileExistAtPath

Huge Difference!!!
Bypassing Sandbox Restrictions --- Case Studies

• Detect whether an app has been installed

DivorceForce  AsthmaMD  Pregnancy+  Sugar Sense
Bypassing Sandbox Restrictions --- Case Studies

• Push notifications:
  • .pushstore file with the bundle identifier as its name will be created in a specific directory
  • (/var/mobile/Library/SpringBoard/PushStore/com.google.Gmail.pushstore for the Gmail app)

• Dynamically registered home screen quick actions:
  • .plist file with the bundle identifier as its name will be created in a specific directory
    (var/mobile/Library/SpringBoard/Application Shortcuts/com.google.Gmail.plist for the Gmail app)

• Top 150 apps in App Store’s “Top Charts” (Aug. 2017):
  • Push notification: 67 (44.7%)
  • dynamically registered home screen quick actions: 44 (31.3%)
Bypassing Sandbox Restrictions --- Case Studies

• Other cases: number of photos/memos

• Generic approach to detect files
Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion
Practical Issues

• App Store Vetting
  • Disguised as an *Audio Player*
  • Passed the vetting

• Power Consumption
  • Device: jailbroken iPhone 7 with iOS 10.1.1
  • 60 min: 5% battery was consumed
Practical Issues --- Cross-device Attack Feasibility

training device: Device A  
iOS 10.1.1

testing device: Device B  
Non-jailbroken iOS 10.2.1
Practical Issues --- Cross-device Attack Feasibility

- Test set: Randomly select 20 third-party apps
- Redo Foreground Apps Experiment

![Diagram showing accuracy results]

- Accuracy (%): 80.5% for Top 1
- Accuracy (%): 91.5% for Top 3

Top N Result: 1, 2, 3, 4, 5
Practical Issues --- Cross-device Attack Feasibility

• Target: *Blockchain Wallet*
Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion
Countermeasures

• Rate Limiting: limit the sampling rate
  • Filter the data and only keep every \((1000/N)th\) data point
  • Re-evaluate the foreground app classification

Implemented in iOS 11.1 for host_statistics64(): 2/s
Countermeasures

- Coarse-grained return values: masking the digits of return values
  - Mask 1/2/3 digits of all 6 features
  - Re-evaluate the foreground app classification

Original: 1234
Mask 1 digit: 1230
Mask 2 digits: 1200
Mask 3 digits: 1000
Countermeasures

- Coarse-grained return values: masking the digits of return values
  - Mask 1/2/3 digits of all 6 features
  - Re-evaluate the foreground app classification

Implemented in iOS 11 for getifaddrs():
Round to 1KB

Accuracy(%)
Countermeasures

- Eliminating the attack vectors
- Runtime detection
- Privacy-preserving statistics reporting
- Removing the `fileExistsAtPath` timing channel

`fileExistsAtPath` timing channel has been eliminated in iOS 11
Outline

1. Side-channel Attack Vectors on iOS
2. Attack 1: Classifying User Activities
3. Attack 2: Detecting Sensitive In-App Activities
4. Attack 3: Bypassing Sandbox Restrictions
5. Practical Issues
6. Countermeasures
7. Conclusion
Conclusion

• First exploration of OS-level side channels on iOS

• Three categories of side-channel attacks

• Proposed countermeasures integrated in iOS and MacOS
THANKS FOR LISTENING

ANY QUESTIONS?

Xiaokuan Zhang
zhang.5840@osu.edu
Detecting Sensitive In-App Activities --- Attack Methods

• Time is short (<0.5s)

• Difference is subtle
Detecting Sensitive In-App Activities --- Attack Methods

• Pattern Matching: compare two multi-dimensional data traces
  • Sample: \( \tilde{X}_t = \{ X_1^t, X_2^t, \cdots, X_l^t \} \), where \( X_i^t = (X_{t_1}^i, X_{t_2}^i, \cdots, X_{t_{n_i}}^i) \)
  • Signature: \( \tilde{S}_t = \{ S_1^t, S_2^t, \cdots, S_l^t \} \)
  • Goal: measure the distance \( d(\tilde{X}_t, \tilde{S}_t) \)
  • Extended DTW (DTW_I): (\( w_k \): normalization factor)

\[
d(\tilde{X}_t, \tilde{S}_t) = \sum_{k=1}^{l} \frac{1}{w_k} \cdot \text{DTW}(\tilde{X}_t^k, \tilde{S}_t^k)
\]
It’s become almost axiomatic that the apps on them are more competition. But researchers continue to expose vulnerabilities in the security protections in macOS and iOS, making it possible to create malware that can be spread through the apps. In addition, it’s also possible to steal data from iCloud passwords to dodgy selfies and more.

The attacks, known as unauthorized access or XARA, expose designs to access critical pieces of data. Apple has struggled to fix the issue released today from Indiana University Bloomington, Peking University and the Georgia Institute of Technology.

Apple is facing another blow to its reputation for security on the iPhone. A flaw in iMessage has been discovered that allows a single message to lock up and potentially crash your handset. And you don’t even have to read the message for it to activate.

The bug itself is relatively easy to explain. When iMessage receives a message with a URL embedded, it will go online and generate a small thumbnail preview of the link. If the metadata is much larger than normally accepted (on the order of hundreds of thousands of characters), then iMessage will lock up the device. The hacker who announced this bug demonstrated it to BuzzFeed News through a poisoned page hosted on Github.
<table>
<thead>
<tr>
<th>Paper</th>
<th>Vector</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen et al., Security’14</td>
<td>/proc/pid/statm</td>
<td>UI inference attacks (stealing login credentials, photos)</td>
</tr>
<tr>
<td>Diao et al., Oakland’16</td>
<td>/proc/interrupts</td>
<td>Interrupt timing analysis (cracking unlock patterns)</td>
</tr>
</tbody>
</table>
Classifying User Activities --- Attack Methods

• Requirements:
  • Combining multiple time series
  • Reducing the dimension

• Major components:
  • Symbolic Aggregate approXimation (SAX) (Keogh et al., 2002)
  • Bag-of-Patterns (BOP) representation (Lin et al., 2009)
  • Support Vector Machine (LibSVM) (Chang et al., 2011)
  {cbb:1, bbc:1, bcc:1, ccc:1, ccb:1, cba:1, baa:1, aaa:1}
Classifying User Activities --- Case Studies

• Top N Accuracy Example

<table>
<thead>
<tr>
<th>Sample</th>
<th>True Class</th>
<th>SVM Prediction (Probability Model)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>
Classifying User Activities --- Case Studies

• Top N Accuracy Example

<table>
<thead>
<tr>
<th>Sample</th>
<th>True Class</th>
<th>SVM Prediction (Probability Model)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
Classifying User Activities --- Case Studies

• Top N Accuracy Example

<table>
<thead>
<tr>
<th>Sample</th>
<th>True Class</th>
<th>SVM Prediction (Probability Model)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Top 1 Accuracy: 3/5 = 60%
Classifying User Activities --- Case Studies

- Top N Accuracy Example

<table>
<thead>
<tr>
<th>Sample</th>
<th>True Class</th>
<th>SVM Prediction (Probability Model)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
Classifying User Activities --- Case Studies

• Top N Accuracy Example

<table>
<thead>
<tr>
<th>Sample</th>
<th>True Class</th>
<th>SVM Prediction (Probability Model)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Top 2 Accuracy: \((3+1)/5 = 80\%\)
Classifying User Activities --- Case Studies

• Top N Accuracy Example

<table>
<thead>
<tr>
<th>Sample</th>
<th>True Class</th>
<th>SVM Prediction (Probability Model)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
Classifying User Activities --- Case Studies

• Top N Accuracy Example

<table>
<thead>
<tr>
<th>Sample</th>
<th>True Class</th>
<th>SVM Prediction (Probability Model)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Top 3 Accuracy: \((2+1+2)/5 = 100\%\)
Detecting Sensitive In-App Activities

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity</th>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:03 PM</td>
<td>Sent Bitcoin</td>
<td>To Bitcoin address</td>
<td>-0.010153</td>
</tr>
<tr>
<td></td>
<td>Bought Bitcoin</td>
<td>Using MasterCard ****4979</td>
<td>0.010153</td>
</tr>
<tr>
<td>APRIL 2017</td>
<td>Sent Bitcoin</td>
<td>To Bitcoin address</td>
<td>-0.01108</td>
</tr>
<tr>
<td></td>
<td>Received Bitcoin</td>
<td>From Bitcoin address</td>
<td>0.011</td>
</tr>
<tr>
<td></td>
<td>Sent Bitcoin</td>
<td>To Bitcoin address</td>
<td>-0.0081068</td>
</tr>
<tr>
<td></td>
<td>Sent Bitcoin</td>
<td>To Bitcoin address</td>
<td>-0.010407</td>
</tr>
<tr>
<td></td>
<td>Sent Bitcoin</td>
<td>To Bitcoin address</td>
<td>-0.0014068</td>
</tr>
<tr>
<td></td>
<td>Bought Bitcoin</td>
<td>Using MasterCard ****4979</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Detecting Sensitive In-App Activities --- Attack Methods

• Identify critical events

• Correlates with public records
Detecting Sensitive In-App Activities

Screen 1: 
- Backup Funds
- Settings
- Addresses
- Merchant Map
- Support

Screen 2: 
- Profile:
  - Wallet ID: X
  - Email: Verified
  - Mobile Number: Verified
- Preferences:
  - Email Notifications
  - SMS Notifications
  - Local Currency: U.S. dollar
  - Bitcoin Unit: Bitcoin
- Security:
  - 2 step Verification: Disabled

Screen 3: 
- Screen showing Bitcoin wallet with address: 0.00888458 BTC
- Options: Send, Transactions, Receive

Screen 4: 
- Confirm Payment
- Details:
  - Amount: 0.00000752 BTC
  - Fee: 0.00000006 BTC
  - Total: 0.00000758 BTC
- Options: Send, Back, Continue
Classifying User Activities --- Case Studies

• Device: jailbroken iPhone 7 with iOS 10.1.1
• Automated using Cycrypt
Why global stat can work?

• iOS itself suspends apps when they run in the background, unless the app specially requests background permissions.

• iOS is relatively quieter than Android, which greatly facilitates side-channel attacks.
Run Background Apps on iOS

• *AUDIO* background mode

• `[NSTimer scheduledTimerWithTimeInterval: target: selector: userInfo: repeats:]`
Detecting Sensitive In-App Activities