Outlines

• Launching Generic Attacks on iOS with Approved 3rd-Party Apps
• Towards Secure Password-based User Authentication
• RFID Security & Privacy

Robert H. Deng
Singapore Management University

ASI@CityU, Hong Kong
2-4 December 2014
Launching Generic Attacks on iOS with Approved 3rd-Party Apps
iOS Security Features

• Unlike other platforms (such as Android):
  – iOS apps have very restricted privileges
    → reason for user to jailbreak
  – Apple app vetting process
  – iTunes App Store and iOS sandbox
iOS Security Effectiveness

• Both app vetting and iOS sandbox mechanisms are black box
  – generally regarded as highly effective
  – **no harmful malware** on non-jailbroken devices has been reported on iTunes App Store [1]
    – only **graywares** were found and then removed [2, 3]

• No official documents for iOS app permissions

Launch Serious Attacks on iOS?

• **Two conditions** for launching such attacks
  – Malicious apps need to pass Apple’s vetting process
  – Attack codes need to bypass iOS sandbox

• Any vulnerabilities in the vetting / iOS sandbox, which allow apps to gain much more privileges on non-jailbroken devices?
Generic Attack Vectors

• The Idea
  - Load the needed framework(s) dynamically
  - Locate the needed classes dynamically
  - Invoke needed private APIs dynamically
  - Obfuscate all strings used in the code (framework name, class name, method name…)

• String obfuscation:
  char* s = “abcdefghijklmnopqrstuvwxyz”
Attacks Implemented

- We implemented 7 attacks, which allow apps to:
  - Crack and retrieve device PIN:
    - 40 s, if the PIN is in birthday format (ddmm/mmdd)
    - 18.2 m to check the whole PIN space \(10^4\)
  - Continuously taking screen snapshots at background:
    - Only works on iPad
- Block incoming calls
- Take photos/videos, post tweets and send SMS/email without user’s awareness
Demos

- PIN-cracking attack
- Snapshot attack
Results and Impacts

• All our apps were embedded with attack codes
  – appeared on the official App Store
  – all attacks work on non-jailbroken iOS devices

• Reported to Apple’s product security team & held conference call with Apple in Oct, 2012.

• Apple fixed PIN-cracking vulnerability in iOS 6.1 (released in Jan 2013)

• Other attacks fixed in iOS 7 when it was formally released (Aug/Sep 2013)

• More details, please refer to paper in ACNS2013
Towards Secure Password-based User Authentication
Password

- Password is the most pervasive means for user authentication
  - Cheap & accurate
  - Support mobility

- The other two basis of user authentication (what you have and what you are) are normally expensive and cumbersome to use
Attacks to Password Systems

**Attacks at the client side**

- Phishing, MITM
- Offline dictionary
  - Well established techniques, e.g., PAKE

**Server database reading**
- Tamper resistant hardware
- 2-server system
Attacks at the Client Side

• Malware, e.g. software keylogger, screen capturer

• Untrusted input device e.g. hardware keylogger

• Keyboard-based eavesdropping e.g. tone, typing pattern and timing based analysis

• Shoulder surfing e.g. hidden camera recording
Leakage-Resilient Password Systems (LRPS)

Assumptions

1. Totally exposed to attacker
   - Software key logger
   - Hidden camera, shoulder surfing
   - Key acoustic analysis, typing pattern analysis

2. Un-aided user

Aided User
Prior efforts on LRPS for unaided humans

EUROCRYPT '91
Matsumoto et al.

EUROCRYPT '95
Wang et al.

ASIACRYPT '01
Hopper & Blum

AVI '06
Wiedenbeck

S&P '06
Weinshall

ACSAC '08
Bai et al.

ACSAC '09
Li et al.

ISC '10
Asghar et al.

Hopper & Blum’s scheme remains intact till today, but imposes significant cognitive workload and memory demand on user.
Why so hard? – capability asymmetry

The adversary

**Advantage:**
Access to protocol transcripts
Computation power
Storage

**Disadvantage:**
Don’t know the password

The user

**Advantage:**
Knowledge of the password

**Disadvantage:**
Limited cognitive computation
impossible to do CPA secure encryption
\( E(\text{secret, challenge}) \)
Limited memory
Inherent Tradeoff?

• None of the existing LRPS systems are both **Secure** and **Usable**

• We studied the **fundamental limitations** of LRPS from the perspectives of
  – Two generic attacks
  – Experimental psychology
User’s password consists of $k$ secret elements out of $n$ elements (the other $n-k$ elements are called decoys).

1. **Challenge**: a window of size $w$ based on a round secret (i.e. a portion of password)

2. **Response** based on the Challenge & round secret

Repeat steps 1 and 2, until the number of correct user responses reaches a **threshold**
Two generic attacks

• **Brute force**
  – Eliminate password candidates that do not lead to correct challenges/responses
  – Effectiveness is *design-independent*
    • Applicable to any LRPS with small password space

• **Statistical analysis**
  – Find out the *most likely* passwords
  – Effectiveness is *design-dependent*
    • Applicable to many LRPSs even with large password space

• They are common knowledge but their power as been *underestimated*
Brute force attack to Undercover

**Undercover** [CHI08, Sasamoto et al.]: *User selects $k = 5$ pictures out of $n = 28$; # of candidate passwords is $C_{28}^5 = 98280$*

At most one secret image appears in a challenge.

Brute force: Given a challenge alone, eliminate all those password candidates who pictures appear more than once in the challenge.

**Challenge**

$P = 1$ (0-indexed)

**Answer** = $(P + r) \mod 5$, where $r$ is a random integer delivered via a secure channel. *Without knowing $r$, the answer tells nothing*
Brute force attack to Undercover

Brute force recovers the exact password after observing 6 sessions
Brute force attack to PAS (Predicate-based Authentication Services)

Password = (32, hello), (13, world)
Challenge index = 2
Round secret = (32, e), (13, o)

Answer = MX
= table[YES, YES][NO, YES]
Brute force attack to round secrets in PAS

The # of round secret in PAS is \((25\times 26)^2 = 422,500\). Given an answer, such as MX, check if a candidate round secret results in the same answer; if not, it’s eliminated. In each round, \(\frac{3}{4}\) of the candidates are eliminated.

Brute force recovers the round secret after observing 1 authentication session.
Statistical Attack to CAS High

CAS High [S&P06, Weinshall]: Password consists of $k = 30$ images out of 80

1. Start from upper-left corner
2. Move down if the current image is a secret image; Otherwise move right
3. Answer = the number associated with the exit
Probabilistic decision tree

- $e \notin \text{secret}$
- $e \in \text{secret}$

Real Decision Path

Consistent Decision Paths

Observed Answer

Other Answers
Score mechanism of probabilistic decision tree

- **Observation**
  - At least one of the consistent decision paths is the correct path
  - Other consistent decision paths are “noises” whose effects cancel out over multiple rounds

- **Probabilities of each decision**
  - $P_1$: $P(e \in \text{secret}) = \frac{k}{n}$
  - $P_0$: $P(e \not\in \text{secret}) = 1 - P_1$

- **Create a table**, each row corresponding to an image; in each round, compute
  - $P(X) = P(<S_1, D_1, D_2, S_2>) = P_1 \cdot P_0 \cdot P_0 \cdot P_1$
  - $P_C$ = Sum of probabilities of all consistent paths
  - Score($S_1$) += $P(X)/P_C$
  - Score($D_1$) -= $P(X)/P_C$
Statistical bias in decision paths

CAS High [S&P06, Weinshall]

43758 possible decision paths in total, with average path length of 14.55

Secret images score significantly higher than decoy images after a sufficient number of observations

Recover the password after observing 65 sessions
Security versus Usability

Security

1. Large password space
2. Large round secret space
3. Uniformly distributed challenges
4. Complex challenges or counting-based challenges

Usability

- Memory
- Computation
- Round Number
- Window Size
- Computation
- Round Number

School of Information Systems
Quantitative evidences from psychology

• Human beings have limitations on cognitive capability and memory
  – These limitations will NOT significantly improved even after repetitive rehearsal

• $12+56-1*5+16+79 = ?$
  – Can you perform this type of computation faster after rehearsal?
  – Can you speed up the calculation by calculating in parallel?
Usability costs in the quantitative analysis framework

• Cognitive workload: $HP(C)$
  – Time that an average human takes to answer a challenge
  – Measured by sum of the reaction time of each atomic operations (e.g., counting, simple arithmetic)
    • Implementation-independent

• Memory demand: $HP(M)$
  – Measured by # of elements memorized $\times$ degree of difficulty of the specific memory retrieval operation
    • Recall is much more difficult than recognition
High security at cost of heavy cognitive demand

More secure

<table>
<thead>
<tr>
<th></th>
<th>k</th>
<th>n</th>
<th>Winsize</th>
<th>Password space</th>
<th>Reported Time/round (sec)</th>
<th>HP (C)/round (sec)</th>
<th>HP (C)/login (sec)</th>
<th>HP (M)</th>
<th>HP Total =M×C (×10^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPN[15]</td>
<td>15</td>
<td>200</td>
<td>200</td>
<td>1.463 × 10^{22}</td>
<td>23.71</td>
<td>33.423</td>
<td>668.45</td>
<td>50.68</td>
<td>338.74</td>
</tr>
<tr>
<td>APW[2]</td>
<td>16</td>
<td>200</td>
<td>200</td>
<td>8.369 × 10^{24}</td>
<td>35.50</td>
<td>57.928</td>
<td>347.57</td>
<td>54.05</td>
<td>187.87</td>
</tr>
<tr>
<td>CAS Low[31]</td>
<td>60</td>
<td>240</td>
<td>20</td>
<td>2.433 × 10^{57}</td>
<td>5.00</td>
<td>6.073</td>
<td>121.46</td>
<td>70.75</td>
<td>85.94</td>
</tr>
<tr>
<td>CAS High[31]</td>
<td>30</td>
<td>80</td>
<td>80</td>
<td>8.871 × 10^{21}</td>
<td>20.00</td>
<td>22.099</td>
<td>220.99</td>
<td>35.38</td>
<td>78.18</td>
</tr>
<tr>
<td>SecHCI[20]</td>
<td>14</td>
<td>140</td>
<td>30</td>
<td>6.510 × 10^{18}</td>
<td>9.00</td>
<td>10.638</td>
<td>212.76</td>
<td>16.51</td>
<td>35.13</td>
</tr>
<tr>
<td>CHC[32]</td>
<td>5</td>
<td>112</td>
<td>83</td>
<td>1.341 × 10^{8}</td>
<td>10.97</td>
<td>9.326</td>
<td>93.26</td>
<td>16.89</td>
<td>15.75</td>
</tr>
</tbody>
</table>

Implications: An LRPS has to incorporate certain trusted “component” in order to be both secure and usable.
KGuard

• Problem
  – Passwords are exposed to the OS and the browser during user authentication

• Goal
  – On-demand protection over user password (keyboard input) against untrusted kernel and browser
KGuard Architecture
KGuard: High level view

- When the user is about to input a password, she invokes the Kguard (i.e., hypervisor) to protect it.
- The Kguard intercepts/fetches password directly from keyboard (not from the driver which is part of the OS); hence prevents keyloggers.
- KGuard encrypts the password using the SSL server’s public key. A dummy password is passed to the OS and then the browser.
- When the browser submits the authentication credential, the dummy password is replaced by the ciphertext of the real password.
- After authentication, the user deactivates the protection.
KGuard – How It Works

KGuard

- works in the hypervisor space
- intercepts password and caches it
- injects the encrypted password into the SSL connection
Guardian

- A tiny hypervisor with around 25K SLOC

- Main features
  - Protects its own integrity and availability against persistent rootkits which may tamper with the hypervisor image in the hard-drive
  - Has a secure user interface for users to securely configure security policies. This is an extension from KGuard
  - Provides two utilities
    - device monitoring: prevent rootkits from misusing peripheral devices (e.g. camera) without the user’s awareness
    - hyper-firewall: prevent rootkits from bypassing firewall polices set by the user
References


RFID Security & Privacy
Introduction

• RFID tags are **low-cost** electronic devices, from which stored info can be collected by an RFID reader **efficiently** and **at a distance** without line of sight

• RFID has found numerous applications, from warehouse inventory control, supermarket checkout counters, e-ticket, to e-passport
RFID Triggered Significant Concerns on Security & Privacy

• **Perfect working condition for attackers**
  – Tags can be read or traced by malicious readers from a distance w/o its owner’s awareness

• **Security**
  – **Tag authentication**: ensure data collected not from fake tag & prevent database pollution
  – **Reader authentication**: prevent unauthorized access to/or tampering with tag data

• **Privacy**
  • **Anonymity**: Confidentiality of the tag identity
  • **Untraceability**: Unlinkability of the tag’s transactions
RFID Security and Privacy at Physical Level

Tags

Reader

Authenticate / Identify

Read / Update

Database

Network
Cryptographic Protocols for RFID Privacy

• Numerous lightweight RFID protocols for low-cost tags have been proposed
  – Use simple operations (XOR, bit inner product, CRC, etc)

• Most of them have been broken

• Need to investigate formal RFID security and privacy models which are fundamental to the design and analysis of robust RFID systems
Assumptions

• $S = \{T_1, \ldots, T_n\}$ - a group of tags
• R/D - Reader/Database have secure connection
• Adversary A has complete control over communications between reader and tags
Canonical RFID Protocol $\pi$

- Shorthand notation: \((c, r, f) \leftarrow \pi(R, T)\)
Game Based Privacy Models

- **Adversary’s capabilities**
  - Modeled by oracles to which adversary can access

- **Adversary’s strategy**
  - Possible combinations of the oracles queries by adversary

- **Adversary’s goal**
  - Successfully tracing tags - winning condition of adversary in the game

- RFID protocol is secure if probability of adversary winning the game is negligible
Oracles

- Interactions between A and protocol parties R and T occur through 4 oracles
  - $O_1$ - Launch(): return a session id $sid$ and the 1st message $c$
  - $O_2$ - SendTag($sid$, $c$, $T$): return $r$, the response of tag $T$
  - $O_3$ - SendReader($sid$, $r$): return $f$, the response of Reader
  - $O_4$ - Corrupt($T$): return the secret information and state of tag $T$
Ind-Privacy: Indistinguishability of two tags [Jules & Weis, PerCom 2007]

Experiment:

- \( \{T_i, T_j\} \leftarrow A_1^{O1,O2,O3,O4}(R, S) \);
- \( b \in \{0, 1\} \);
- If \( b = 0 \) then \( T_c = T_i \), else \( T_c = T_j \);
- \( S' = S - \{T_i, T_j\} \);
- \( b' \leftarrow A_2^{O1,O2,O3,O4}(R, S', T_c) \).

- Advantage of adversary \( A = |\Pr[b'=b]-1/2| \)

\( A_1 \) not allowed to query \( O_4 \) on \( T_i \) and \( T_j \)

\( A_2 \) not allowed to query \( O_4 \) on \( T_c \)
Ind-Privacy

- Indistinguishability of two tags
- The idea is intuitively appealing
- But the model is difficult to apply directly in proving a given protocol is ind-private
- To our knowledge, no protocol has been proven directly to satisfy the model
Unp*-Privacy (Ha, Moon, Zhou & Ha, *ESORICS 2008*; Lai, Deng, Li, *ACNS 2010*)

**Experiment:**

1. \( T_c \leftarrow A_1^{O_1,O_2,O_3,O_4}(R, S); \)
2. \( b \in \{0, 1\}; \)
3. **When** \( A_2 \) **makes queries to** \( O_1, O_2, O_3 \) **on** \( T_c \)
   - If \( b = 0 \), return oracles’ responses
   - Else (\( b = 1 \))
     - return \( c \in_R C \) if query \( O_1 \)
     - return \( r \in_R R \) if query \( O_2 \)
     - Return \( f \in_R F \) if query \( O_3 \)
4. \( b' \leftarrow A_3 \)

- The advantage of adversary \( A = |Pr[b'=b]-1/2| \)
• Unp*-Privacy Model (Ha, Moon, Zhou & Ha, ESORICS’08; Ma, Li, Deng & Li CCS’09, Lai, Deng, Li, ACNS’10)
  – Adversary can not tell protocol messages \( \pi(R, T) = (c, r, f) \) from random messages
  – Too restrictive, precludes PKC techniques
Relationships

- Unp*-Privacy is equivalent to PRF
- Unp*-Privacy is stronger than Ind-privacy but probably too strong
- ZK-Pri is also stronger than Ind-Privacy
RFID Security and Privacy at Network Level

Tags

Authenticate / Identify

Reader

Read / Update

Database

Network
RFID Network Level

• Incorporating RFID data collected at physical level with other information (e.g., location, time, business steps) to form RFID events

• Storing and sharing these events among networked parties

• A typical example is the EPCglobal Network
  
  – EPCglobal is an organization for creating worldwide standard for Electronic Product Code (EPC) technology, including RFID and use of the Internet to share RFID data
  
  – Over 3000 participants in standardization process
EPCGlobal Network – Discovery Service (EPCDS) and Information Service (EPCIS)

1. Publish: (epc1, A, T1)

2. Publish: (epc1, B, T2)

... 

EPCDS

(epc1, A, T1)
(epc1, B, T2)
...

EPCIS A
At Partner A

Tags are transported from Partner A to Partner B

EPCIS B
At Partner B

Authorized User

4. Reply: (epc1, A, T1)

5. Query: (epc1, T1)

6. Reply: (epc1, T1)

3. Query: epc1
EPCglobal Network

EPC-DS: database of epc event indexing records

<table>
<thead>
<tr>
<th>EPC</th>
<th>EPCIS</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>epc1</td>
<td>A</td>
<td>T1</td>
</tr>
<tr>
<td>epc1</td>
<td>B</td>
<td>T2</td>
</tr>
</tbody>
</table>

EPC-IS: database of EPC events

RFID physical level

Party A

Party B
## EPC Events

| What        | • EPC number  
|            | • Manufacturing Data (lot, batch, expiration date)  
|            | • Transactional Data (PO, Shipment, Invoice)  |
| Where      | • Location (fixed or moving)  |
| When       | • Event Time  
|            | • Record Time  |
| Why        | • Business Process Step - e.g. Receiving, Shipping  
|            | • Product State - e.g. Saleable, Active, In Transit  
|            | • Current Conditions - e.g. Temperature  |
RFID Security & Privacy at Network Level

• **Security**
  
  – Authentication and access control at EPCIS and EPCDS
  – Data integrity (e.g., detection of false EPC event injection)
  – DoS protection

• **Privacy**
  
  – Compared to clandestine scanning at the physical level, unauthorized tracking at the network level could be more harmful as adversary can obtain tracking information on a global scale and without physical presence
Summary & Future Directions

- Relatively good understanding of security and privacy issues for RFID system at physical level; but crypto protocols which satisfy formal models are too expensive for low cost tags.

- Symmetric key crypto implementation in tags
  - Smallest 128-bit AES requires 3400 gates
  - Hash based on block cipher PRESENT with 128-bit output requires 4000 gates
  - SHA-1 requires 5500 gates
  - 5 to 10 cent tag has 1000 to 10,000 gate counts

- Very few research efforts on RFID security & privacy at the network level.
References


• Deng, Li, Yung, Zhao: A New Framework for RFID Privacy. *The 15th European Symposium on Research in Computer Security (ESORICS)*, 2010


Thank You!