Breaking into the iCloud Keychain

Vladimir Katalov
ElcomSoft Co.Ltd.
Moscow, Russia
[Some of] our customers
What do we want to hack today?

1. All user’s passwords
2. Credit card data
What’s inside the smartphone?

*(tip: almost everything)*

- Contacts & calendars
- Call logs and text messages
- Emails and chats
- **Account and application passwords**
- **Web and Wi-Fi passwords**
- **Credit card data**
- Documents, settings and databases
- Web history & searches
- Pictures and videos
- Geolocation history, routes and places
- 3rd party app data
- Cached internet data
- System and application logs
- Social network activities
Data acquisition methods | Cloud acquisition pros and cons

- **JTAG/chip-off**
  - No test access port on many devices
  - Full disk encryption

- **Physical**
  - Limited compatibility
  - Data may be encrypted

- **Logical**
  - Limited compatibility
  - Bypassing screen lock is needed

- **Cloud**
  - Limited set of data // *oh, really? 😊*
  - Need credentials
  - Legal problems

- **Problems**
  - Different platforms (Apple, Google, Microsoft)
  - Many vendor-specific clouds
  - 3rd party cloud services
  - Credentials needed (password or token)

- **Profits**
  - No physical access needed
  - May be performed silently

- **Backup**
  - No standard way to get
  - Might not be available
  - Almost all data from device

- **Sync**
  - Limited set of data
  - Most critical real-time data
  - Synced across all devices

- **Storage**
  - Only files/documents
  - Easy to access
Cloud services: backups [iCloud]

- Full device backups are sometimes available
- 3rd party application data is usually not available
- Passwords are additionally encrypted with hardware-specific key
- Daily backups (in best case, until forced from the device)
- Backups cannot be forced remotely
- 3rd party software is needed
- Almost no way to manage
- Slow access, long download
- Account might be locked due to ‘suspicious activity’
Cloud services: synced data [iCloud]

- Contacts
- **Call log**
  - Messages (SMS/iMessage, CallKit-compatible apps)
- Calendars
- Mail (only cloud-based)
- Internet activities (visited sites, searches)
- Media files (photos, videos)
- Gaming data
- **Passwords**
- Health data
- **Credit cards**

- Other
- ApplePay
- Home devices
- Wallet
- Maps (searches, bookmarks, routes)
- Books
- News, weather
- **Location data**
More iCloud data

- Account information
- iCloud storage information
- Contact information (billing/shipping address, emails, credit cards (last 4 digits)
- Connected devices
- Customer service records
- iTunes (purchase/download transactions and connections, update/re-download connections, Match connections, gift cards)
- Retail and online store transactions
- Mail logs
- Family sharing data
- iMessage and FaceTime metadata
- Deleted data?
Apple keychains

- **iOS keychain**
  - Local (encrypted backup)
  - Local (not encrypted backup)
  - iCloud
    
    View (iOS 10): Settings | Safari | Passwords, AutoFill  
    View (iOS 11): Settings | Accounts & Passwords | App & Website Passwords  
    Protection: it depends  
    Decrypt/export: no way (3rd party software only)

- **OS X (macOS) keychain**
  
  View: Keychain utility (one by one)  
  Protection: password (by default, same as log on)  
  Decrypt/export: 3rd party software only

- **iCloud keychain**
  
  View: Only when/if synced with local device  
  Protection: well, strong 😊  
  Decrypt/export: ?
Backup vs iCloud keychains

<table>
<thead>
<tr>
<th></th>
<th>Backup</th>
<th>iCloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wi-Fi</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Web sites</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Credit cards</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>App-specific</td>
<td>+</td>
<td><em>It depends</em></td>
</tr>
<tr>
<td>AirPlay/AirPort</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Encryption keys &amp; tokens</td>
<td>+</td>
<td><em>It depends</em></td>
</tr>
<tr>
<td>Autocomplete</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

*Keychain in iCloud backups have most data encrypted with device-specific key*
iOS keychain – passwords (Wi-Fi, email, web form)

- **AirPort**
  - **AP name**: AirPort
  - **Service**: AirPort
  - **Account**: AP name
  - **Data**: AP password
  - **Access Group**: apple
  - **Creation Date**: 20121231120800.529226Z
  - **Modification Date**: 20121231120800.529226Z
  - **Protection Class**: CLASS: 7

- **accounts.google.com**
  - **Email**: accounts.google.com
  - **Account**: email
  - **Data**: password
  - **Protocol**: HTTPS
  - **Authentication Type**: form
  - **Description**: Web form password
  - **Access Group**: com.apple.cfnetwork
  - **Creation Date**: 20150705071047.78112Z
  - **Modification Date**: 20150805133813.889686Z
  - **Protection Class**: CLASS: 6

- **imap.gmail.com**
  - **Email**: imap.gmail.com
  - **Account**: email
  - **Data**: password
  - **Protocol**: IMAP
  - **Port**: 143
  - **Access Group**: apple
  - **Creation Date**: 20121231124745.097385Z
  - **Modification Date**: 20121231124745.097385Z
  - **Protection Class**: CLASS: 7
iOS keychain (credit card data)

- `<Name>SafariCreditCardEntries (BBA00CB1-9DFA-4964-B6B8-3F155D88D794)</Name>`
- `<Service>SafariCreditCardEntries</Service>`
- `<Account>BBA00CB1-9DFA-4964-B6B8-3F155D88D794</Account>`
- `<Data>`
  - `<Dictionary>`
    - `<CardholderName>NAME</CardholderName>`
    - `<ExpirationDate>DATE</ExpirationDate>`
    - `<CardNameUIString>Visa</CardNameUIString>`
    - `<CardNumber>NUMBER</CardNumber>`
  - `</Dictionary>`
- `</Data>`
- `<Comment>This keychain item is used by Safari to automatically fill credit card information in web forms.</Comment>`
- `<Access Group>com.apple.safari.credit-cards</Access Group>`
- `<Creation Date>20131016100432.283795Z</Creation Date>`
- `<Modification Date>20150826181627.118539Z</Modification Date>`
- `<Label>Safari Credit Card Entry: Visa</Label>`
- `<Protection Class>CLASS: 6</Protection Class>`
iOS [backup] keychain protection classes

- kSecAttrAccessibleAfterFirstUnlock (7)
  The data in the keychain item cannot be accessed after a restart until the device has been unlocked once by the user.

- kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly (10)
  The data in the keychain item cannot be accessed after a restart until the device has been unlocked once by the user.

- kSecAttrAccessibleAlways (8)
  The data in the keychain item can always be accessed regardless of whether the device is locked.

- kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly
  The data in the keychain can only be accessed when the device is unlocked. Only available if a passcode is set on the device.

- kSecAttrAccessibleAlwaysThisDeviceOnly (11)
  The data in the keychain item can always be accessed regardless of whether the device is locked.

- kSecAttrAccessibleWhenUnlocked (6)
  The data in the keychain item can be accessed only while the device is unlocked by the user.

- kSecAttrAccessibleWhenUnlockedThisDeviceOnly (9)
  The data in the keychain item can be accessed only while the device is unlocked by the user.

- **ThisDeviceOnly**: encrypted using device-specific hardware key (can be extracted from 32-bit devices only)
- All others: in password-protected local backups, encrypted with the key derived from backup password
iTunes backup password breaking

- Get manifest.plist
- Get BackupKeyBag
- Check password
  - iOS 3
    - `pbkdf2_sha1(2,000)`
  - iOS 4 to 10.1 (but 10.0)
    - Same as above, but 10,000 iterations
  - iOS 10.0
    - Same as above works
      - *Single sha256 hash is also stored*
  - iOS 10.2+
    - `pbkdf2_sha256(10,000,000)`
    - `pbkdf2_sha1(10,000)`
- Unwrap AES key from KeyBag
- Decrypt keychain (+other files?)

Hashes are salted, so no rainbow tables ☹️
macOS keychain
iCloud data protection


- Most of the data: A minimum of 128-bit AES encryption
- iCloud Keychain: Uses 256-bit AES encryption to store and transmit passwords and credit card information. Also uses elliptic curve asymmetric cryptography and key wrapping.

- Key is stored along with the data (except just the iCloud keychain)!

- Notification to email when the data is accessed
- Account might be blocked due to suspicious activity (new!)
- Two-step verification (legacy, not recommended)

  **Two-factor authentication**
  - Immediate push notification to all trusted devices
  - Have to allow access
  - Security code
    - As push notification
    - By SMS to trusted phone number
    - Generated by trusted device

- Workaround for 2FA: use authentication token from the device (iPhone/iPad/iPod), PC or Mac
iCloud sign-in

Your Apple ID is being used to sign in to a new device near Amsterdam, North Holland.

Do Not Allow  Allow
About iCloud keychain
Set up iCloud keychain – no 2FA

To use iCloud Keychain on this iPhone, approve it from another device that uses iCloud Keychain or use your iCloud Security Code.

Approve with Security Code

iCloud Keychain
Keep your passwords and credit card information up to date across the devices you approve. Your information is encrypted and cannot be read by Apple.
Set up 2FA

Add an additional layer of security to your account to protect the photos, documents and other data you have stored with Apple.

RESCUE EMAIL ADDRESS

v*****@elcomsoft.com  Verified

If you have forgotten your password or security questions, this email address can be used to help you reset them.

Enter the code sent to *******20.

Did not get a verification code?

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>ABC</td>
<td>DEF</td>
</tr>
<tr>
<td>5</td>
<td>JKL</td>
<td>MNO</td>
</tr>
<tr>
<td>6</td>
<td>GHI</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TUV</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 6 6 5 1 3
Set up iCloud keychain –2FA

Enter Device Password
Some of your data stored in iCloud is protected by the password you use to unlock "VK-IMAC".
Your password is encrypted and cannot be read by Apple.

Forgot Device Password?
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?
space
return

Choose Device
Your account is protected by your device passcodes.

ENTER PASSCODE FOR
- Vladimir Katalov's iPad Mini 4
- iPad mini 4
- VK-IMAC
- Mac mini
- iPhone 7
- iPhone 7

Forgot all passcodes?
iOS 11 and 2FA

Two-Factor Authentication
Two-factor authentication is the best way to keep your account secure. It can protect your account even if someone has found out your password.

When signing in, you will verify your identity using one of your devices or your phone number.

Turn On
Learn More
Not Now
iCloud sync modes

**Recovery**: recovery from keychain backup/storage in the iCloud

*com.apple.sbd3 (Secure Backup Daemon)*

Keep backup of keychain records, and copying to new devices (when there are new trusted ones)

**Sync**: real-time syncing across cloud and devices

*com.apple.security.cloudkeychainproxy3*

Support for “trusted circle”, adding new devices to it etc
• iCloud circle of trust

• **iOS Security Guide:**

• **Keychain syncing**
  • Circle of trust
  • Public key: syncing identity (specific to device)
  • Private key (elliptical P256), derived from iCloud password
  • Each synced item is encrypted specifically for the device (cannot be decrypted by other devices)
  • Only items with `kSecAttrSynchronizable` are synced

• **Keychain recovery**
  • Secure escrow service (**optional**)
  • No 2FA: iCloud security code is needed (+SMS)
  • No 2FA, no iCSC: recovery is not possible
  • 2FA: device passcode is needed
  • Hardware Security Module (WTF is that? 😊)
iCloud keychain recovery mode

3: key version (GCM or CBC algorithm; GCM here).
6: record protection class (KeyBag #6 here)
0x48: wrapped key size
Next: encrypted key data
iCloud keychain recovery protection (no 2FA)

**iCSC - iCloud Security code**

**No iCSC**

Sync mode only. Keychain records are not stored in the iCloud and cannot be recovered if all trusted devices are lost. Access is possible only through push notification to the trusted device. *The most safe/secure config? ;)*

**iCSC is set**

- Push notification to trusted device (as above)
- iCSC plus code from SMS (6 digits)

*Note:* *iCSC is not stored anywhere in the cloud, just its hash (in Escrow). Three options are available:*

- Simple (4 or 6 digits, depends on iOS version)
- Complex (any symbols, up to 32)
- Device-generated/random (24 symbols)
iCloud keychain recovery protection (2FA)

For every device, separate record is created (at EscrowProxy):

\texttt{com.apple.icdp.<deviceHash>}

Contents: BackupBagPassword (randomly generated)

Usage: RFC6637 to encrypt keys from iCloud Keychain Keybags
Escrow proxy architecture (1)

- **Escrow proxy**

  - SRP (Secure Remote Password) protocol
  - Safe from MITM
  - Does not need password to be transferred at all (even hash)
  - Does not keep password on server
Escrow proxy architecture (2)

- **Cloud Keychain records of interest at EscrowProxy**

  - `com.apple.securebackup`: keep BackupBagPassword from Keybag, where iCloudKeychain is stored for ‘full restore’
  
  - `com.apple.icdp.<deviceHash>`: BackupBagPassword from iCloudKeychain individual records from given devices, stored for partial recovery
Escrow proxy architecture (3)

No 2FA (iCSC) and 2FA (Device Passcode):

- Client generates random 25-symbol KeyBagKey
- PBKDF2(SHA256, 10000) to generate iCSC/passcode hash
- KeyBagKey is encrypted with AES-CBC using hash as a key
- Encrypted KeyBagKey is stored in EscrowProxy

Note: if ‘random’ option is selected as iCSC, then it is not hashed, and saved ‘as is’ It is further used for encrypting KeyBag with set of keys for iCloud Keychain.
## Escrow proxy API

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>/get_club_cert</td>
<td>Returns certificate, associated with account</td>
</tr>
<tr>
<td>/enroll</td>
<td>Add new secure record</td>
</tr>
<tr>
<td>/get_records</td>
<td>Get list of stored records</td>
</tr>
<tr>
<td>/get_sms_targets</td>
<td>Get phone number, associated with account</td>
</tr>
<tr>
<td>/generate_sms_challenge</td>
<td>Sends approval code via sms to associated number</td>
</tr>
<tr>
<td>/srp_init</td>
<td>Initializes authentication via SRP-6a protocol</td>
</tr>
<tr>
<td>/recover</td>
<td>SRP authentication finalization. returns secure records on success</td>
</tr>
<tr>
<td>/update_record</td>
<td>Updates records information associated with account</td>
</tr>
</tbody>
</table>
Escrow proxy: ‘public’ records

- Info on key used for protection
- Number of failed retries
- Device data (model, version, password strength)
- List of keys for KeyBag decryption
- Protected Storage Services list
SRP protocol (v6)

iCSC - iCloud Secure Code

H – SHA256
N, g – 2048-bit generator of the multiplicative group (RFC 5054)

The user enroll password verifier and salt to EscrowCache. EscrowCache stores password verifier and salt.

\[\text{<salt>} = \text{random}()\]
\[x = \text{SHA}(<\text{salt}>) | \text{SHA(<dsid>) | ":" | <iCSC>})\]
\[<\text{password verifier}> = v = g^x \mod N\]

- If `com.apple.securebackup` record exists, that means iCloud Security Code is set. Otherwise, EscrowProxy contains `com.apple.icdp.record.hash_of_device` records, so iCloud Keychain can be synced when one of device passwords is provided.
Escrow proxy – access tokens

- No 2FA, iCloud Security Code: MME token is enough; validation uses SMS to trusted number set in account

- How to obtain: same as for backups, synced data, iCloud Photo Library etc

- 2FA, device passcode: PET (Password Equivalent Token); TTL=5 minutes

- How to obtain: pass GSA authentication (to approve short-time access from the given device); new in macOS 10.11

<table>
<thead>
<tr>
<th>Record name</th>
<th>Authentication Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.apple.securebackup</td>
<td>MME + SMS</td>
</tr>
<tr>
<td>com.apple.icdp.record.hash_of_device</td>
<td>PET</td>
</tr>
<tr>
<td>com.apple.protectedcloudstorage</td>
<td>MME</td>
</tr>
</tbody>
</table>

Authentication type for access of Escrow record
Keychain is sync mode

In sync mode, KeyBag may contain as full records in recovery mode (BackupKeyBag, com.apple.securebackup.record) or *tombs*, unique for every domain (HomeKit, Wi-Fi etc)
Tombs

- Keybag & metadata (ASN.1 format)
- Keychain: records for the given domain, encrypted with Keybag
- Wrapped Key (for every RecordID): Keybag key wrapped with RFC 6637

To decrypt

- get tombs from com.apple.sbd
- find all RecordIDs
- get BackupBagPassword for the given RecordID, using passcode of the device
- unwrap KeyBag key
- decrypt keys from KeyBag
- Decrypt Keychain records
Other components and alternative approaches

**GSA (GrandSlam Authentication)**

- gsa.apple.com
- based on SRP protocol
- introduced in macOS 10.10 (basic)
- improved in macOS 10.11

**AnisietteData**

- MachineID + OTP
- MachineID (60 bytes): unique for device
- OTP (24 bytes): random; refreshed every 90 seconds
- code is hardly obfuscated
- implemented in Apple Private API

**Continuation token**

- obtained through GSA
- means to get tokens for other services
- no need to keep Apple ID and password on device
- can be used to get *updated* tokens with short TTL
- for further requests: use **AlternateDSID & Continuation token** instead of AppleID & password
No 2FA

- Apple ID
- Password
- iCloud security code
- SMS to trusted number

2FA

- Apple ID
- Password
  
  no need to pass 2FA on trusted Desktop
- Passcode of enrolled device
Conclusions / risks

• Sync and recovery: different approaches
• Trusted circle: not hard to get in, but leaves traces
• Both sync and recovery can be used (mixed)
• Need to have credentials
• Need to have trusted device
  ...or SMS
• Need to know iCSC
  ...or device passcode
• Legacy 2SV: forget it
• With 2FA, keychain is always stored in iCloud
• No 2FA, no iCSC: most safe from TLA?

• Get *Continuation token (+machine ID)* to obtain full access without anything else!
• *...implementation is still relatively secure 😊*
Wait, one more thing...

- iCloud Keychain contains more data than officially documented: not just passwords, but also tokens (e.g. to 2FA-protected social network accounts and secure messengers), including ones not available even with full physical acquisition

- iCloud Keychain is being activated right when you enable 2FA (or even always exist??), though contains only system keys, not user data

- iCloud Keychain contains encryption keys used to lock some new iCloud data (iOS 11)

What else do you hide from us, Apple? :)

Thanks! Questions?