Beyond files forensic
OWADE cloud based forensic

Elie Bursztein  Stanford University
Ivan Fontarensky  Cassidian
Matthieu Martin  Stanford University
Jean Michel Picod  Cassidian
The world is moving to the cloud
2.7 millions photos are uploaded to Facebook every 20 minutes
100 millions new files are saved on Dropbox every day
Data are moving to multiple services

Hard drive
Data are moving to multiple services

Hard drive → emails
Data are moving to multiple services

Hard drive → emails

Cloud
Data are moving to multiple services

- Hard drive
- Cloud
- emails
- Webmail
Data are moving to multiple services

- Hard drive
- Cloud
- emails
- contacts
- Webmail

Wednesday, August 3, 2011
Data are moving to multiple services

- Hard drive
- Cloud
- Emails
- Contacts
- Webmail
- Social sites
Data are moving to multiple services

Hard drive → emails → contacts → photos → Cloud → Webmail → Social sites

E. Bursztein, I. Fontarensky, J.M. Picod, M. Martin
Beyond files recovery: OWADE cloud based forensic
Wednesday, August 3, 2011
http://owade.org
Data are moving to multiple services

- Hard drive
- emails
- contacts
- photos
- Photo sites
- Cloud
- Webmail
- Social sites

Beyond files recovery: OWADE cloud based forensic

http://owade.org
Data are moving to multiple services

Hard drive → emails → contacts → photos → Cloud → Webmail → Social sites → Photo sites
Impact on the forensic field

• There are more data which are harder to reach
• Dealing with cloud data force us to reinvent forensic
Let’s do cloud forensics
What is cloud forensics?
Facebook credentials as a use case

Facebook

Wednesday, August 3, 2011
Facebook credentials as a use case
Facebook credentials as a use case

DPAPI master-key

IE
DPAPI Blob

Facebook

DPAPI blob-key

credentials
Facebook credentials as a use case

SAM (hash) → Windows User Password ← DPAPI master-key

IE DPAPI Blob → credentials

Facebook

DPAPI blob-key
Facebook credentials as a use case

Registry → Syskey → SAM (hash) → Windows User Password → DPAPI master-key → DPAPI blob-key → IE DPAPI Blob → Facebook credentials
Facebook credentials as a use case

Getting Facebook credentials require to bypass 4 layers of encryption
Show you how to **bypass the encryption layers** and get the data you want
Introducing OWADE

• Dedicated to cloud forensics
• Decrypt / recovers
  • DPAPI secrets
  • Browsers history and websites credentials
  • Instant messaging creds
  • Wifi data
• Free and open-source

http://owade.org
Beyond files recovery: OWADE cloud based forensic
Beyond files recovery: OWADE cloud based forensic

http://owade.org
Beyond files recovery: OWADE cloud based forensic

http://owade.org

disk
Beyond files recovery: OWADE cloud based forensic

http://owade.org
OWADE overview

- disk
- disk image
- Registry
OWADE overview

- disk
- disk image
- Registry
- Files

Beyond files recovery: OWADE cloud based forensic

http://owade.org
Beyond files recovery: OWADE cloud based forensic

http://owade.org
OWADE overview

- disk
- disk image
- Registry
- Windows credentials
- WiFi info
- Files
OWADE overview

- Disk
- Disk image
- Registry
- Windows credentials
- WiFi info
- Files
- Hardware info
OWADE overview

- Disk
- Disk image
- Registry
- WiFi info
- Files
- Hardware info
- Windows credentials
- Credentials and data

Beyond files recovery: OWADE cloud based forensic

http://owade.org
Beyond files recovery: OWADE cloud based forensic

OWADE overview

disk  disk image
Registry
Files
WiFi info
Hardware info
Credentials and data
Cloud data

E. Bursztein, I. Fontaresky, J.M. Picod, M. Martin

Wednesday, August 3, 2011

http://owade.org
Outline

• File base forensics refresher
Outline

- File base forensics refresher
- The Windows crypto eco-system
Outline

• File base forensics refresher
• The Windows crypto eco-system
• Wifi data and Geo-location
Outline

- File base forensics refresher
- The Windows crypto eco-system
- Wifi data and Geo-location
- Recovering browser data
Outline

• File base forensics refresher
• The Windows crypto eco-system
• Wifi data and Geo-location
• Recovering browser data
• Recovering instant messaging data
Outline

• File base forensics refresher
• The Windows crypto eco-system
• Wifi data and Geo-location
• Recovering browser data
• Recovering instant messaging data
• Acquiring cloud data
Outline

• File base forensics refresher
• The Windows crypto eco-system
• Wifi data and Geo-location
• Recovering browser data
• Recovering instant messaging data
• Acquiring cloud data
• Demo
File based forensic refresher
<table>
<thead>
<tr>
<th>Type of file</th>
<th>how to recover it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>copy</td>
</tr>
<tr>
<td>In the trash</td>
<td>undelete utility</td>
</tr>
<tr>
<td>Deleted</td>
<td>file carving</td>
</tr>
<tr>
<td>Wiped</td>
<td>call the NSA :)</td>
</tr>
</tbody>
</table>
Windows registry

- .dat files
- Hardware information
- Softwares installed with their versions and serials
- Windows credentials (encrypted)
Some Registry Information Extracted

Expand All | Contract All
- Web Analyze
- Program Analyze
- Get User Password
- Get User Environment Detail
- Get Software Environment Detail
- Get System Environment Detail
- Get Software Detail
- Get User Configuration Detail
- Get OS Detail
- Get Hardware Detail
- FDC
- USBSTOR
  - Friendly Name 1: Generic USB MS Reader USB Device
  - Friendly Name 0: Generic USB CF Reader USB Device
  - Friendly Name 3: Disk drive
  - Friendly Name 2: Generic USB SD Reader USB Device
  - Friendly Name 5: HP v100w USB Device
  - Friendly Name 4: HP v100w USB Device
- SW
- ACPI
  - Friendly Name 1: Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz
  - Friendly Name 0: Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz
  - Friendly Name 3: Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz
  - Friendly Name 2: Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz
  - Friendly Name 5: Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz
  - Friendly Name 4: Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz
  - Friendly Name 7: Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz
  - Friendly Name 6: Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz
- PCI
- SCSI
- IDE
- PCIDE
- DISPLAY
- Files Statistics

E. Bursztein, I. Fontarensky, J.M. Picod, M. Martin
Beyond files recovery: OWADE cloud based forensic
http://owade.org

Wednesday, August 3, 2011
Windows crypto
Why do we care about Windows crypto?
The Windows crypto eco-system

Crypto API
The Windows crypto eco-system

Crypto API

SAM
The Windows crypto eco-system

Crypto API

DPAPI

SAM
The Windows crypto eco-system

Crypto API

SAM

DPAPI

Credential Manager

Crypto API

SAM

DPAPI

Credential Manager
Windows Crypto API

- Basic cryptographic blocks
  - Cipher: 3DES, AES
  - Hash functions: SHA-1, SHA256, HMAC
  - PKI: public keys and certificates (X.509)
• Store Windows user credentials
• Located in the registry
• Encrypted with the SYSKEY
• Passwords are hashed
Two hash functions used

- **LM hash function (NT, 2K, XP, VISTA)** weak
- **NTLM (XP, Vista, 7)**

Passwords are **not salted**
LM hash weakness

- Use only upper-case
- Hash password in chunk of 7 characters

\[\text{mypassword} \rightarrow \text{LMHash(MYPASSW)} + \text{LMHash(ORD)}\]

Password key-space: \(69^7\) (at most)
Rainbow Tables

- Pre-compute all the possible passwords
- Time-Memory trade-off
- Rainbow tables of all the LM hash are available
How OWADE Works

- Extract Usernames and password hashes
- LM hashes available?
  - use John/Rainbow tables to get the pass in uppercase
  - use NTLM hashes to find the password cases
- Try to crack the NTLM using John/Rainbow table
Windows Password recovered

Quick Access
Passwords
Linkedin

Datos
Slot: 0
Offset: 63
Size: 78107967
Type: NTFS (0x07)

Expand All | Contract All
- WebAnalyze
- ProgramAnalyze
- GetUserPassword
  - DPAPI_SYSTEM: AQAANj5MkJwI9I4KEdMR7QpYK4YjsnIDPcxe6spCj0AppsF5oc8=
    - Administrator
    - Guest
    - Ashoe
      - id: 1003
      - name: Ashoe
      - ntvhash: 31dh8e0d16ae931b73c59d7e0c089c0
      - lmhash: aad3b435b51404eeaad3b435b51404ee
      - Impass: empty
      - UpdatustUser
      - HelpAssistant
        - id: 1000
        - name: HelpAssistant
        - ntahash: 56c41a7826d75b655976315817291c
        - lmhash: 63429969051b0748db81e98a2be98a
        - Impass: Unknown
- SUPPORT_388945a0
- GetUserEnvironmentDetail
- GetSoftwareEnvironmentDetail
- GetSystemEnvironmentDetail
- GetSoftwareDetail
- GetUserConfigurationDetail
- GetOSDetail
- GetHardwareDetail
- FilesStatistics
If the password is too strong we can't recover it
but we can **still decrypt** DPAPI secret (sometime)
The Data Protection API

• Ensure that encrypted data can’t be decrypted without knowing the user Windows password

• Blackbox crypto API for developers:
  • Encrypt data → DPAPI blob
  • Decrypt DPAPI blob → data

• Main point: tie the encryption to the user password
DPAPI derivation scheme

User → SHA1(password) → pre-key
DPAPI derivation scheme

User → SHA1(password) → pre-key → master-key
DPAPI derivation scheme

User

SHA1(password) → pre-key

→ master-key

→ blob key
DPAPI derivation scheme

User → SHA1(password) → pre-key → master-key → blob key → DPAPI blob
Beyond files recovery: OWADE cloud based forensic

DPAPI derivation scheme

User → SHA1(password) → pre-key → master-key → blob key → DPAPI blob

http://owade.org
DPAPI Blob structure

```c
struct wincrypt_datablob {
    DWORD     cbProviders,
    GUID      pbProviders[cbProviders],
    DWORD     cbMasterkeys,
    GUID      pbMasterkeys[cbMasterkeys],
    DWORD     dwFlags,
    DWORD     cbDescription,
    BYTE      pbDescription[cbDescription],
    ALG_ID    algCipher,
    DWORD     cbKey,
    DWORD     cbData,
    BYTE      pbData[cbData],
    DWORD     dwUnknown,
    ALG_ID    algHash,
    DWORD     dwHashSize,
    DWORD     cbSalt,
    BYTE      pbSalt[cbSalt],
    DWORD     cbCipher,
    BYTE      pbCipher[cbCipher],
    DWORD     cbCrc,
    BYTE      pbCrc[cbCrc]
} ;
```
struct wincrypt_masterkey_masterkeybloc
{
    DWORD dwRevision,
    BYTE pbSalt[16],
    DWORD dwRounds,
    ALG_ID algMAC,
    ALG_ID algCipher,
    BYTE pbEncrypted[]
};
DPAPI blob
DPAPI blob → Master-key GUID → Master key → pre-key
DPAPI blob → Master-key GUID → Master key → pre-key → SHA1(password) → User

Master key
Beyond files recovery: OWADE cloud based forensic

DPAPI blob → Master-key GUID

Master key → Cipher + key → pre-key → Master key

User → SHA1(password)
DPAPI blob → Master-key GUID → Master key

Cipher + key → pre-key

Master key → blob key

SHA1(password) → User
Beyond files recovery: OWADE cloud based forensic

DPAPI blob → Master-key GUID

Master key → SHA1(password)

Cipher + key → pre-key

IV + Salt → Master key

Master key → blob key

User

http://owade.org
Beyond files recovery: OWADE cloud based forensic

DPAPI blob → Master-key GUID

Master key:
- Cipher + key
- IV + Salt
- Additional entropy

User

Software

http://owade.org
• If we can’t crack the password we need its SHA1

• This SHA1 is stored in the hibernate file

• OWADE uses Moonsols to recover it
• Software can supply an additional entropy
  • Act as a “key” (needed for decryption)
  • Force us to understand how it is generated for each software
  • Can be used to tie data to a specific machine (i.e. Netbios name)
Credential Manager

• Built on top of DPAPI
• Handle transparently the encryption and storage of sensitive data
• Used by Windows, Live Messenger, Remote desktop...
# Credstore type of credentials

<table>
<thead>
<tr>
<th>Type of credential</th>
<th>Encryption</th>
<th>Example of application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic password</td>
<td>DPAPI + fixed string</td>
<td>Live messenger HTTP auth (IE)</td>
</tr>
<tr>
<td>Domain password</td>
<td>In clear</td>
<td>Netbios</td>
</tr>
<tr>
<td>Domain certificate</td>
<td>Hash of certificate</td>
<td>Certificate</td>
</tr>
<tr>
<td>Domain visible password</td>
<td>DPAPI + fixed string</td>
<td>Remote access .NET passport</td>
</tr>
</tbody>
</table>
WiFi data
Wifi data

- Info stored for each access point
  - Mac address (BSSID)
  - Key (encrypted)
  - Last time of access
- Wifi data are stored in
  - Registry (XP)
  - XML file and Registry (Vista/7)
Decompressing WiFi password

- Encrypted with DPAPI
- Access point shared among users
  - Encrypted with the System account
  - But the system account has no password...

What is my DPAPI key ???
Decrypting WiFi password

• Use a LSASecret as DPAPI key

• Array of credentials
  • HelpAssistant password in clear
  • DPAPI_SYSTEM

• “Encrypted”
Where are you?

- We’ve recovered access point keys but where are they?
• We’ve recovered access point *keys* but where are they?

There is an app for that!
Location-Aware Browsing

Firefox can tell websites where you’re located so you can find info that’s more relevant and more useful. It’s about making the Web smarter – and is done in a way that totally respects your privacy. Give it a try!

Frequently Asked Questions

+ What is Location-Aware Browsing?
+ How does it work?
+ How accurate are the locations?
+ What information is being sent, and to whom? How is my privacy protected?
+ Am I being tracked as I browse the web?
HTML5 Geo-location protocol

Location-Aware Browsing

Firefox can tell where you are, and you can find information useful. It’s about doing it in a way that respects privacy. Give it a try.

Frequently Asked Questions

+ What is Location-Aware Browsing?
+ How does it work?
+ How accurate are the locations?
HTML5 Geo-location protocol
Google Wi-Fi Data Collection Angers European Officials

Brennon Slattery, PC World  May 17, 2010 7:08 am
European officials are still miffed over Google's "accidental" Wi-Fi data collection and seek an in-depth investigation that may lead to harsh penalties for the search engine giant.

It was revealed that Google's Street View cars were collecting more than images and coordinates for its sophisticated GPS site. As much as 600GB of data from Wi-Fi networks -- in more than 30 countries -- has been snagged in Google's fishnet.
Beyond files recovery: OWADE cloud based forensic

E. Bursztein, I. Fontarensky, J.M. Picod, M. Martin

http://owade.org

Nothing is ever easy

• Google started to restrict queries in June
• So we started to look for other API
• Live service
• “Documented” in the Windows mobile MSDN
• After sniffing the traffic:
  • Use a big SOAP request
  • Does not check any ID fields
  • Allows to supply one MAC

```xml
<GetLocationUsingFingerprint xmlns="http://inference.location.live.com">
  <RequestHeader>
    <Timestamp>2011-02-15T16:22:47.0000968-05:00</Timestamp>
    <ApplicationId>e1e71f6b-2149-45f3-b298-a20XXXXX5017</ApplicationId>
    <TrackingId>21BF9AD6-CFD3-46B2-B042-EE90XXXXXX</TrackingId>
    <DeviceProfile ClientGuid="0fc571be-4622-4ce0-b04e-XXXXXXeb1a222" Platform="Windows7" DeviceType="PC"
OSVersion="7600.16695.amd64fre.win7_gdr.101026-1503"
LFVersion="9.0.8080.16413" ExtendedDeviceInfo="" />
    <Authorization />
  </RequestHeader>
  <BeaconFingerprint>
    <Detections>
      <Wifi7 BssId="00:BA:DC:0F:FE:00" rssi="-25" />
    </Detections>
  </BeaconFingerprint>
</GetLocationUsingFingerprint>
```
Beyond files recovery: OWADE cloud based forensic

E. Bursztein, I. Fontarensky, J.M. Picod, M. Martin

http://owade.org

Blog post and demo released!

From Information to Intelligence
Dealing with information in the digital age

Using the Microsoft Geolocalization API to retrace where a Windows laptop has been
July 29, 2011 | Privacy
4 Comments and 93 Reactions

About the author
Ellie Bursztein
I am a security researcher at Stanford University. This blog is about security, and more broadly about web technologies.

Search...
Search

Subscribe and follow us

Latest Security Report
Report from the security frontlines July 18 th 24 th
July 26, 2011

13,188 people visited this site

13,963 Visits
13,188 Unique Visitors
51,805 Pageviews
3.71 Pages/Visit
00:02:04 Avg. Time on Site
22.94% Bounce Rate
94.13% New Visits
Just fixed

• Fixed last weekend
• No longer return location for a single address

Microsoft Privacy & Safety
Microsoft's Approach to Helping Protect Privacy and Safety

Microsoft Makes Change to Geographic Location Positioning Service
Microsoft Privacy Team 1 Aug 2011 11:58 AM 0

Updated 9:14 A.M. 8/2/2011
Microsoft released a change to its geographic location positioning service on July 30, 2011, which addresses an issue highlighted in Elie Bursztein's blog on July 29, 2011. This change adds improved filtering to validate each request so that the service will no longer return an inferred position when a single Media Access Control address is submitted. Microsoft is keenly aware of the sensitivity around all privacy issues, especially those surrounding geolocation.

Microsoft's privacy and security team has been in contact with Elie and we will continue the ongoing dialog with experts in the privacy field to improve our service offerings. We thank Elie, Matthieu Martin from Stanford University, Jean Michael Picod and Ivan Fontarensky from Cassidian for working with us on this issue.

Microsoft's commitment to privacy means that not only will we seek to build privacy into products, but we'll also engage with key stakeholders in government, industry, academia and public interest groups to develop more effective privacy and data protection measures. We will continue to update our service with improvements that benefit the consumer in both positioning accuracy as well as individual privacy.

Reid Kuhn is a Partner Group Program Manager on the Windows Phone engineering team at Microsoft
Privacy
• Fixed last weekend
• No longer return location for a single address
Geo-location API restrictions

Requires 2 MAC close from each other

The MAC and IP location need to be “close”

Requires multiples MAC addresses

see http://elie.im/blog/ for more information
WiFi Information Extracted By OWDE
Browsers
Firefox > 3.4

- **Passwords**
  - Location: `signons.sqlite`
  - Encryption: `3DES + Master password`

- **History**
  - URLs: `places.sqlite`
  - Forms fields: `formhistory.sqlite`
Decrypting Firefox password
Decrypting Firefox password

User → pass
Decrypting Firefox password

User → pass → Global salt ← key3.db
Decrypting Firefox password

User → pass → user key: HMAC-SHA1(salt, pass) → Global salt → key3.db
Decrypting Firefox password

User

pass

user key: HMAC-SHA1 (salt, pass)

Global salt

key3.db

encrypted key + key salt

key3.db
Decrypting Firefox password

User → pass → Global salt → key

user key: HMAC-SHA1(salt, pass)

encrypted key + key salt

master key: 3DES(userkey, enckey)
Decrypting Firefox password

User → pass → user key: HMAC-SHA1 (salt, pass) → Global salt → master key: 3DES(userkey, enckey) → encrypted key + key salt → encrypted pass → signon.sqlite

key3.db
Decrypting Firefox password

User

User

pass

Global salt

user key: HMAC-SHA1 (salt, pass)

key3.db

encrypted key + key salt

master key: 3DES (userkey, enckey)

key3.db

encrypted pass

Site password: 3DES (master key, enc pass)

signon.sqlite
Beyond files recovery: OWADE cloud based forensic

E. Bursztein, I. Fontarensky, J.M. Picod, M. Martin

http://owade.org
Beyond files recovery: OWADE cloud based forensic

E. Bursztein, I. Fontarensky, J.M. Picod, M. Martin

http://owade.org
Beyond files recovery: OWADE cloud based forensic

E. Bursztein, I. Fontarensky, J.M. Picod, M. Martin

http://owade.org

How about a nice kindle?

Wednesday, August 3, 2011
Every form field is recorded

<table>
<thead>
<tr>
<th>id</th>
<th>fieldname</th>
<th>value</th>
<th>timesUsed</th>
<th>firstUsed</th>
<th>lastUsed</th>
<th>quid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>email</td>
<td><a href="mailto:testblackhat@devnull.com">testblackhat@devnull.com</a></td>
<td>1311823018543000</td>
<td>1311823018543000</td>
<td>CTwZ4J59TYySue7B</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>enterAddressFirstName</td>
<td>Allan Smith</td>
<td>1311823235859000</td>
<td>1311823235859000</td>
<td>FXPxpcwKuRq5TonB+</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>enterAddressAddressLine1</td>
<td>42 my street</td>
<td>1311823235861000</td>
<td>1311823235861000</td>
<td>84y6gVgFt4QmGoIQ</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>enterAddressCity</td>
<td>San Francisco</td>
<td>1311823235861000</td>
<td>1311823235861000</td>
<td>84y6gVgFt4QmGoIQ</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>enterAddressStateOrRegion</td>
<td>CA</td>
<td>1311823235861000</td>
<td>1311823235861000</td>
<td>84y6gVgFt4QmGoIQ</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>enterAddressPostalCode</td>
<td>94302</td>
<td>1311823235862000</td>
<td>1311823235862000</td>
<td>et3sZGGQ26+2mjq</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>enterAddressPhoneNumber</td>
<td>666-666-6666</td>
<td>1311823235862000</td>
<td>1311823235862000</td>
<td>84y6gVgFt4QmGoIQ</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>enterAddressAddressLine1</td>
<td>my street</td>
<td>1311823259915000</td>
<td>1311823259915000</td>
<td>h8ULVNgTFuglDeDY</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>searchbar-history</td>
<td></td>
<td>1311823282116000</td>
<td>1311823282116000</td>
<td>poNbcyN2SsydiObN</td>
<td></td>
</tr>
</tbody>
</table>
Configuring a Linksys?

Security Mode: WPA2/WPA Mixed Mode
Passphrase: thisismypwakey

Security Mode: WPA2/WPA Mixed Mode
Passphrase: thisismypwakey

Save Settings  Cancel Changes
Again the key is recorded
Form history leak a lot of information

- Shipping address
- Wifi key
- Credit card information
- Email
- Search history
To tell the browser to not record a field use the tag

\texttt{autocomplete=“off”}
• Passwords
  • Location: registry
• Encryption: DPAPI + URL as salt
• History
  • URLs: Index.dat
Decrypting Internet Explorer passwords
Decrypting Internet Explorer passwords

Registry

SHA1(URL)

Beyond files recovery: OWADE cloud based forensic

http://owade.org
Decrypting Internet Explorer passwords

Registry → SHA1(URL) → URL → URL List
Decrypting Internet Explorer passwords

Registry → SHA1(URL) → URL (dpapi entropy) → URL List
Decrypting Internet Explorer passwords

Registry → SHA1 (URL) → Key → URL (dpapi entropy) → URL List

DPAPI Blob → Registry
Decrypting Internet Explorer passwords

- **Registry** → SHA1(URL) → Key

- **URL** (dpapi entropy) → URL List

- **Site password** → DPAPI Blob → Registry
Maximizing our recovery

- Build a list of URL from others browsers and files
- Use a list of known login URLs
• Passwords
• Location: Login Data (sqlite)
• Encryption: DPAPI

• History
• URLs: History (sqlite)
• Forms fields: Web Data (sqlite)
• Passwords
  • Location: keychain.plist (Property list format)
  • Encryption: DPAPI + fixed string as entropy

• History
  • URLs: History.plist
  • Forms fields: Form Value.plist
Browsers takeaway

- Internet Explorer is the most secure.
  - If you don’t know the URL you can’t recover the credentials
- Firefox is the worst
  - Passwords encryption not tied to the Windows user password (bug open for a while)
  - Login are encrypted in signons.sqlite not in formhistory.sqlite
Private mode

- Most bugs are fixed
- Requires to be creative
  - SSL OCSP requests
  - File carving
- Potential techniques
  - Analyze the hibernate file

See: http://ly.tl/p16 for more information on private mode
The browsers histories aggregated

<table>
<thead>
<tr>
<th>Website</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.com</td>
<td>375</td>
</tr>
<tr>
<td>live.com</td>
<td>41</td>
</tr>
<tr>
<td>facebook.com</td>
<td>35</td>
</tr>
<tr>
<td>neufailk.fr</td>
<td>31</td>
</tr>
<tr>
<td>skype.com</td>
<td>30</td>
</tr>
<tr>
<td>microsoft.com</td>
<td>28</td>
</tr>
<tr>
<td>aol.com</td>
<td>26</td>
</tr>
<tr>
<td>youtube.com</td>
<td>25</td>
</tr>
<tr>
<td>ashe.fr</td>
<td>21</td>
</tr>
<tr>
<td>twitter.com</td>
<td>20</td>
</tr>
<tr>
<td>doubleclick.net</td>
<td>16</td>
</tr>
<tr>
<td>gmodules.com</td>
<td>16</td>
</tr>
<tr>
<td>msn.com</td>
<td>15</td>
</tr>
<tr>
<td>clubic.com</td>
<td>11</td>
</tr>
<tr>
<td>rot.com</td>
<td>10</td>
</tr>
<tr>
<td>ie9enhanced.com</td>
<td>10</td>
</tr>
<tr>
<td>apple.com</td>
<td>9</td>
</tr>
<tr>
<td>hotmail.com</td>
<td>9</td>
</tr>
<tr>
<td>accelacom.com</td>
<td>8</td>
</tr>
<tr>
<td>bing.com</td>
<td>8</td>
</tr>
<tr>
<td>fbcdn.net</td>
<td>8</td>
</tr>
<tr>
<td>steampowered.com</td>
<td>8</td>
</tr>
<tr>
<td>aim.com</td>
<td>8</td>
</tr>
<tr>
<td>wixrs.com</td>
<td>8</td>
</tr>
<tr>
<td>adtm.com</td>
<td>7</td>
</tr>
<tr>
<td>sourceforge.net</td>
<td>7</td>
</tr>
<tr>
<td>cnet.com</td>
<td>7</td>
</tr>
<tr>
<td>mydigitallife.info</td>
<td>6</td>
</tr>
<tr>
<td>touslesdrivers.com</td>
<td>5</td>
</tr>
</tbody>
</table>
Instant messaging
- Encryption custom
- Difficulty extreme
- Location registry + config.xml
Decrypting Skype passwords
Decrypting Skype passwords

Registry → DPAPI Blob → pre-key
Decrypting Skype passwords

Registry → DPAPI Blob → pre-key

AES key: SHA1(pre-key)
Decrypting Skype passwords

Registry → DPAPI Blob → pre-key → AES key: SHA1(pre-key) → encrypted credential

config.xml
Decrypting Skype passwords

- DPAPI Blob
- pre-key
- AES key: SHA1(pre-key)
- encrypted credential

Login:
- pass cracking
- MD5(login\nskyp\npassword)

config.xml

E. Bursztein, I. Fontarensky, J.M. Picod, M. Martin
Decrypting Skype passwords

There is a John the ripper patch for that

- DPAPI Blob
- pre-key
- AES key: SHA1(pre-key)
- encrypted credential
- MD5(login\nskyper\npassword)
- Login

config.xml
Google Talk

- Encryption
  DPAPI + custom (salt)

- Difficulty
  Hard

- Location
  registry
Salt derivation algorithm overview
Salt derivation algorithm overview

String: 0xBA0DA71D
Salt derivation algorithm overview

String: 0xBA0DA71D → Windows account name

Registry

Wednesday, August 3, 2011
Salt derivation algorithm overview

String: 0xBA0DA71D → \(\bigcirc\) → Windows account name

Registry

Wednesday, August 3, 2011
Salt derivation algorithm overview

String: 0xBA0DA71D → Windows account name

Registry

Windows account name → computer Netbios name

computer Netbios name → Registry
Salt derivation algorithm overview

String: 0xBA0DA71D → Windows account name

Windows account name → computer Netbios name

computer Netbios name → Registry

Registry

Registry
Salt derivation algorithm overview

String: 0xBA0DA71D → Windows account name

computer Netbios name

DPAPI Blob

Registry

Registry

Registry
Salt derivation algorithm overview

String: 0xBA0DA71D → Windows account name

→ computer Netbios name

→ DPAPI Blob

→ Key
Microsoft Messenger

- Encryption
  DPAPI or Credstore

- Difficulty
  Medium

- Location
  version dependent
### Windows Messenger by version

<table>
<thead>
<tr>
<th>Version</th>
<th>Storage</th>
<th>encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Registry</td>
<td>Base64 encoded</td>
</tr>
<tr>
<td>6</td>
<td>Credstore</td>
<td>Credstore</td>
</tr>
<tr>
<td>7</td>
<td>Registry x2</td>
<td>DPAPI x 2</td>
</tr>
<tr>
<td>Live</td>
<td>Credstore</td>
<td>Credstore</td>
</tr>
</tbody>
</table>

---

Beyond files recovery: OWADE cloud based forensic

E. Bursztein, I. Fontarensky, J.M. Picod, M. Martin

http://owade.org

Wednesday, August 3, 2011
• Encryption
  **DES**
  key: substr(login . “dummykey”, 8)

• Difficulty
  easy

• Location
  config.xml
• **Encryption**
  XOR
  key: 9

• **Difficulty**
  trivial

• **Location**
  user.config
• Encryption
  Base 64 + XOR
  key: fixed string

• Difficulty
  trivial

• Location
  user.config
• Encryption
  Clear aka encryt-what?

• Difficulty
  none

• Location
  account.xml
• Encryption
  Clear aka encryt-what?

• Difficulty
  none

• Location
  account.xml
• Encryption
  Custom

• Difficulty
difficult (offline)

• Location
registry
Paltalk encryption algorithm

VolumeSerial Number

01234567
Beyond files recovery: OWADE cloud based forensic

E. Bursztein, I. Fontarensky, J.M. Picod, M. Martin

http://owade.org

Paltalk encryption algorithm

VolumeSerial Number
01234567

Paltalk account name
myusername

Registry

Wednesday, August 3, 2011
Beyond files recovery: OWADE cloud based forensic

Paltalk encryption algorithm

VolumeSerial Number: 01234567
Paltalk account name: myusername

Registry

m0y1u2s3e4r5n6a7me x 3
Paltalk encryption algorithm

VolumeSerial Number: 01234567
Paltalk account name: myusername

m0y1u2s3e4r5n6a7me × 3

encrypted password: yyyy yyyy yyyy yyyy

Registry

Wednesday, August 3, 2011
Paltalk encryption algorithm

VolumeSerial Number

01234567

Paltalk account name

myusername

m0y1u2s3e4r5n6a7me x 3

encrypted password

yyy z yyy z yyy z yyy z

\( c_i: \text{yyy} z_i - \text{asciiCode}(S-\text{BOX}_{n-i}) \)
Paltalk encryption algorithm

VolumeSerial Number: 01234567
Paltalk account name: myusername

\text{moylu23e5n6ame} \times 3

encrypted password: \text{yyyyy yyyzy yyyzy}

\text{ci: yyyzi - asciiCode(S-BOX}_{n-i})
Messenger take away

• If your Skype password is strong we can’t recover it
• Gtalk and Paltalk are the only ones to use computer information
• 3rd party software are the least secure
<table>
<thead>
<tr>
<th>Application</th>
<th>Login</th>
<th>Password</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome</td>
<td>owade</td>
<td>rootroot</td>
<td>ashe.fr</td>
</tr>
<tr>
<td>Chrome</td>
<td>project.owade</td>
<td>rootroot</td>
<td>google.com</td>
</tr>
<tr>
<td>Safari</td>
<td>owade</td>
<td>rootroot</td>
<td>ashe.fr</td>
</tr>
<tr>
<td>Trillian</td>
<td>project.owade</td>
<td>rootroot</td>
<td></td>
</tr>
<tr>
<td>GTalk</td>
<td><a href="mailto:project.owade@gmail.com">project.owade@gmail.com</a></td>
<td>rootroot</td>
<td></td>
</tr>
</tbody>
</table>

**Most used**

**Passwords**

- rootroot

**Usernames**

- owade
- project.owade
Cloud based forensic
Cloud modules

- Leverage the credentials and history extracted to get cloud-data
- Might be legal (or not)
- Only LinkedIn currently (more modules almost ready)
OWADE status

• Alpha stage
  • Tested on Ubuntu against XP windows

• Roadmap
  • Stabilizing the code
  • modularize the code so you write your own modules
  • More cloud probes: Facebook, Flickr, Emails...
  • Windows Vista and 7 integration
Conclusion

- People moving to the cloud means more data that is harder to get.
- Forensics needs to evolve to cope with this.
- OWADE is the first tool dedicated to cloud forensic.
  - Decrypt the 4 major browsers data.
  - Decrypt Instant messaging credentials.
  - Open-source.
Thank you!

Please remember to complete your feedback form :)
Download OWADE
http://owade.org

Follow-us on Twitter
@elie, @projectowade

Donate to OWADE to support it!