
# Cyber Piatto del giorno

*centro antiveleni Bologna 051-333333*

# Il cuoco


Francesco Picasso

Reality Net System Solutions

@dfirfpi

blog.digital-forensics.it



HackInBo

# Salsa *cyber*

- Punto di osservazione: DFIR, Network Security
- Personale esperienza e *ricerca* ***notturna*** condita in salsa cyber

***Cyber…***

- Nessun intento di spiegare il termine né la sua storia
- *Percepito* dal sottoscritto come la ***matrice*** di Gibsoniana memoria
- **Interconnessione**

# Antipasto!

Internet delle cozze

*avariate*

# Internet degli oggetti

*«che ogni cosa abbia il suo IP!»*

Quanto è bello interconnettere il frigo, la tv, la macchina (OMG), la casa…

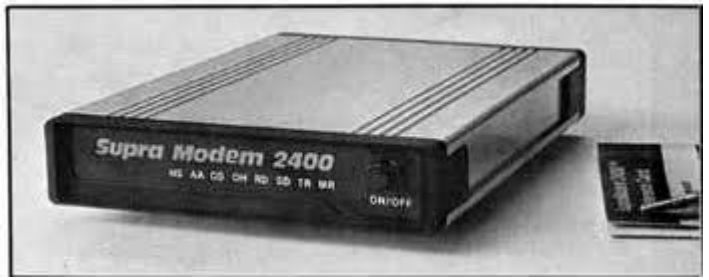
Stimati circa 50 miliardi di dispositivi collegati fra di loro nel 2020.

Tutto connesso, tutto *smart*… tutto sicuro!

Un esempio **odierno** di oggetto tipicamente collegato H24 con il resto del mondo, non intelligente e che non sia un cellulare?

# Campagna di sensibilizzazione

*Sono il tuo modem, non abbandonarmi nello sgabuzzino!*




- Hai mai cambiato la password di gestione del tuo modem/router?
- Hai mai aggiornato il suo firmware? (Quante volte figliuolo?)
- Hai mai verificato/modificato la sua configurazione?
- (*advanced*) lo monitori?



HACKINBO

# Piatto del giorno

# Windows User Password

*(cracking)*
*(rivisitato)*



HACKINBO

# WUP (Windows User Password): perché?

È facile bypassarla…

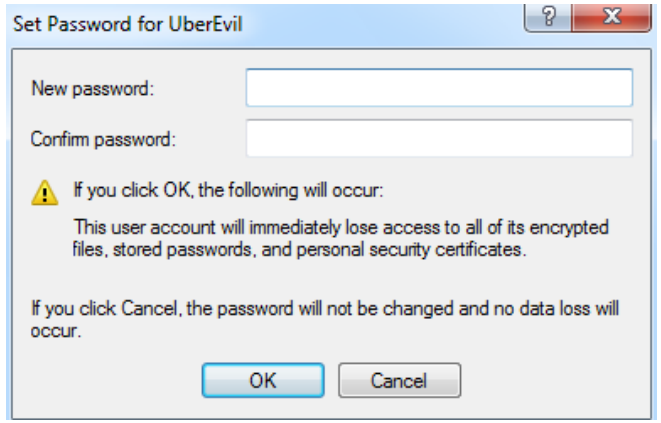
Ntpasswd, Koon-boot, etc.

Ma esistono elementi che **dipendono** da essa

DPAPI (**D**ata **P**rotection **API**)

EFS (**E**ncryption **F**ile **S**ystem)

…




O magari servirebbe conoscerla per ragioni di *intelligence*

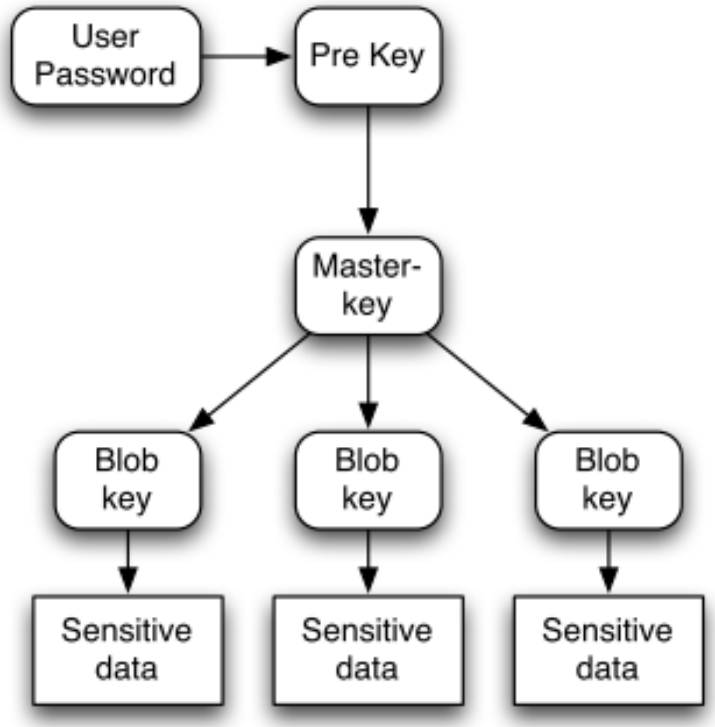


HACKINBO

# DPAPI: inter-inizio

Schema crittografico che cifra/decifra in modo trasparente dati.

La chiave di ingresso al fantastico mondo DPAPI è derivata in modo sicuro dalla password dell'utente.




Due sole primitive

**CryptProtectData**

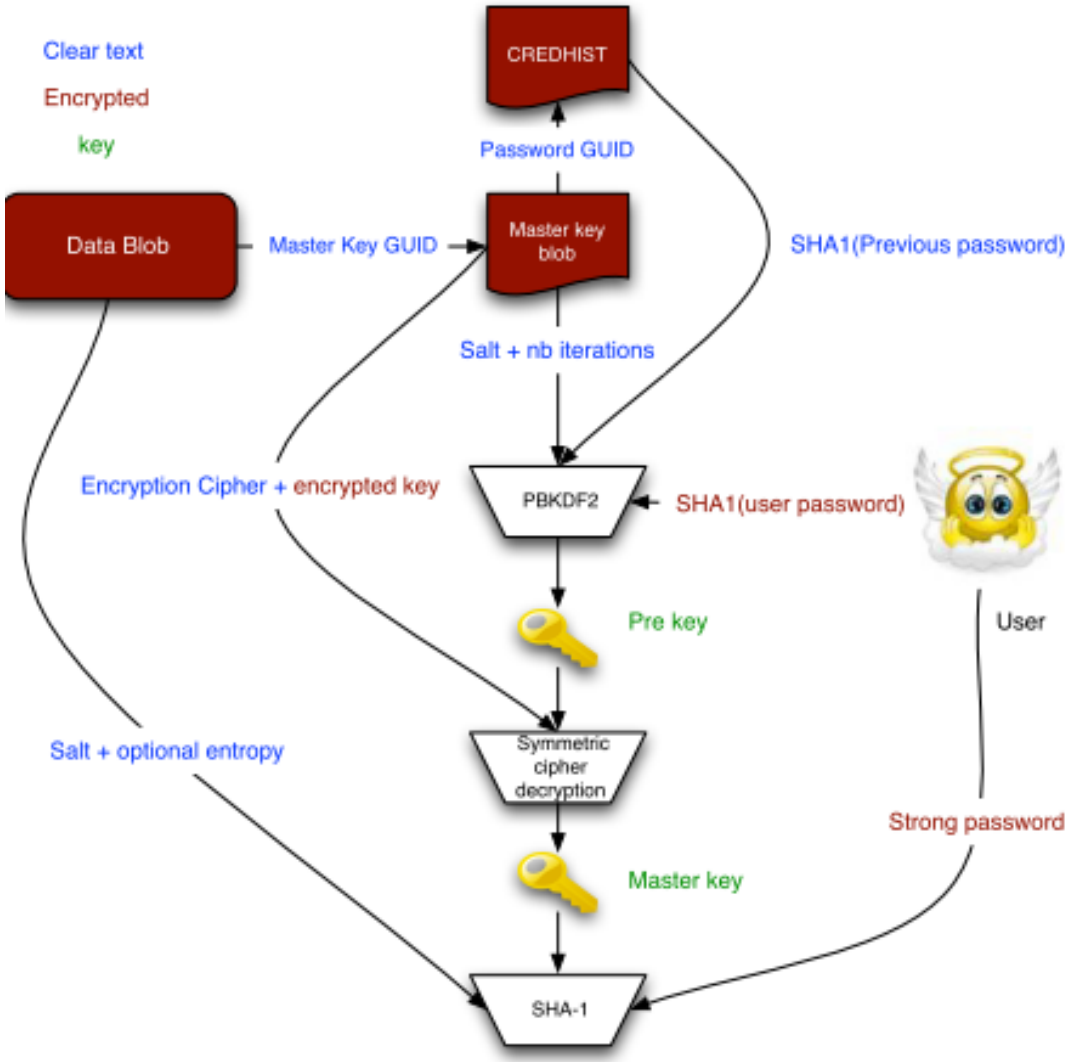
**CryptUnprotectData**

«Una» struttura dati *opaca*, **DPAPI BLOB**



HACKINBO

# DPAPI: inter-mezzo




«*Recovering Windows Secrets and EFS Certificates Offline*»
E. Burzstein, J.M. Picod 2010

```
BOOL WINAPI CryptUnprotectData (
DATA_BLOB *pDataIn,
LPCWSTR *ppszDataDescr,
DATA_BLOB *pOptionalEntropy,
PVOID pvReserved,
CRYPTPROTECT_PROMPTSTRUCT
*pPromptStruct,
DWORD dwFlags,
DATA_BLOB *pDataOut)
```

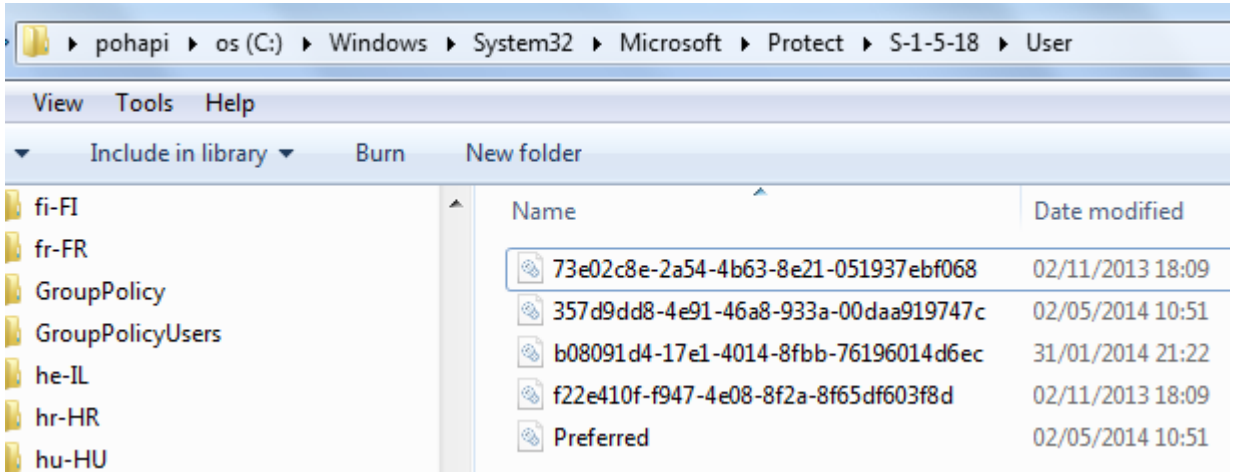
- **MasterKey** rinnovata ogni 3 mesi
- **Cambio password utente**?

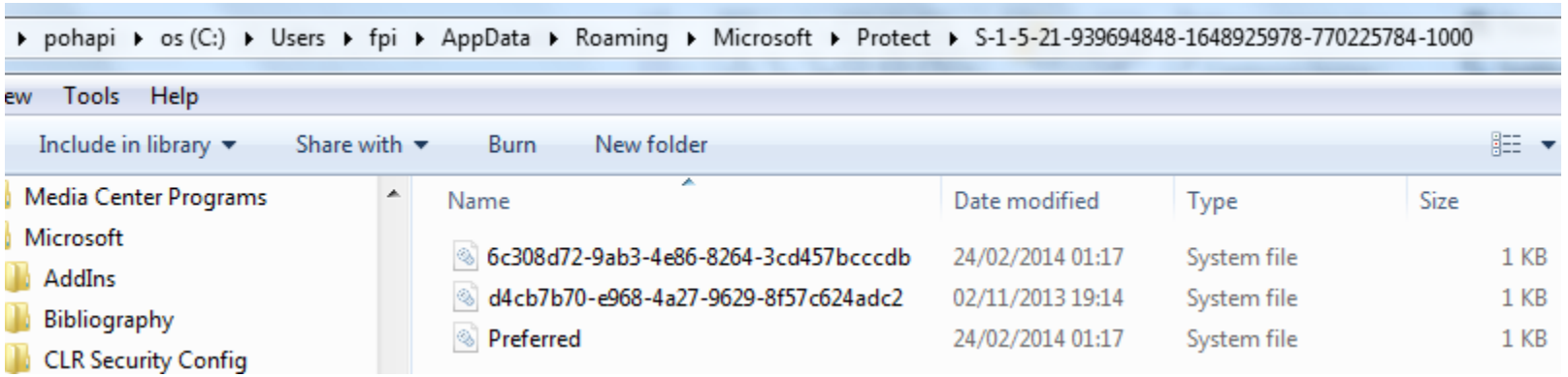


HACKINBO

# DPAPI: inter-fine

- EFS
- Windows Mail
- Internet Explorer
- Gtalk
- Dropbox
- MSN
- Skype
- Outlook
- Wifi
- Windows Vault
- …






HACKINBO

# WUP cracking: **ingredienti**

**Ingredienti**

NTLM hash (LM hash?)

**creddump** (moyix, python oss)

**Rainbow Tables** (dizionario, forza brutTa..)

**password cracker** (qualsiasi tool oss)

***Hands On***



HACKINBO

# Anti WUP cracking

**Rainbow Table**

Sono sufficienti **13,759,005,997,841,642** (circa 2^53,6) elementi?

(ntlm_mixalpha-numeric#1-9)

Una password «decente» è insuperabile…
Che ne dite di
«*D0mani!VadoAdHackInBo?PerchèMiHaInvitatoMari0#»

Fonte RainbowCrack



HACKINBO

# WUP cracking secondo **Chuck**




HACKINBO

# WUP cracking? No grazie, ho smesso

Nuova ricetta! Ingredienti

**hiberfil.sys**

**volatility**
(che gran pezzo di software)

**mimikatz** volatility plugin
(by @dfirfpi)

***Hands On***



HACKINBO

# WUP cracking? No grazie, ho smesso

In presenza di hiberfil o RAM dump

~~possono esistere più hiberfil, per altro…~~

Allora è possibile ottenere le credenziali in chiaro (no cracking) degli utenti *logged in*.

Altrimenti… *roll back* to WUP cracking (sig).

## Piatto del giorno

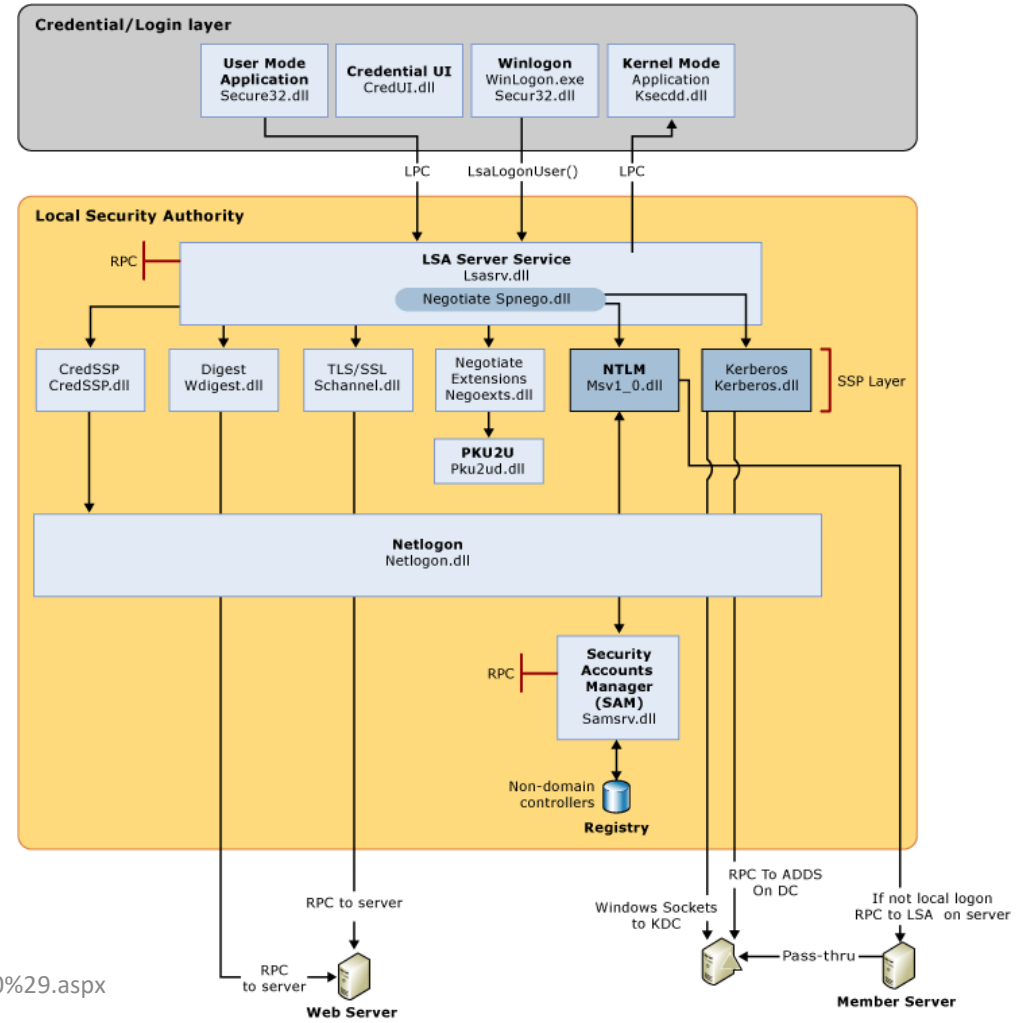
# Windows User Password
# No More Cracking

*ricetta*



HACKINBO

# Ricetta (ovvero dietro le quinte)


http://technet.microsoft.com/en-us/library/dn169016%28v=ws.10%29.aspx



HACKINBO

# PaCChi di autenticazione

Una volta immesse le credenziali, esse sono memorizzate dagli **auth-pkg…**

Perché anche le password?

| | |
|---|---|
| msv1_0 | • NTLM<br>• No password, HASH |
| tspkg | • Terminal Server<br>• password |
| wdigest | • Autenticazione HTTP<br>• password |
| livessp | • Windows live (winotto – filotto!)<br>• password |
| kerberos | • Kerberos (*what else?*)<br>• password (uh, strano…) |

# GentilKiwi

Diamo a Cesare quel che è di Cesare (anche se gallico…)

Benjamin DELPY a.k.a. GentilKiwi
http://blog.gentilkiwi.com/mimikatz
https://github.com/gentilkiwi/mimikatz

## Windows SSO e PTH



HACKINBO

# SSO & PTH secondo Russinovich

## Pass-the-Hash == Single-Sign On

- Pass-the-hash is the use of a saved credential or authenticator
  - It exists solely to support single-sign on (SSO)
  - If you want SSO, you are exposed to PTH
- In other words:
  - If you want SSO, pass-the-hash cannot be “fixed”
  - This is not a “Windows problem”
- There are two types of pass-the-hash:
  - Credential reuse: using the saved credential on the system on which it was saved
  - Credential theft: taking the saved credential to another system and using it from there

Microsoft

2

#RSAC

RSACONFERENCE2014



HACKINBO

# SSO & wdigest

La presenza di **hash** (e, quindi, PTH) e/o **pwd** (e, quindi, credenziali in chiaro) dipende da quale protocollo è stato SSO-ato (che l'italiano mi perdoni).

Wdigest == HTTP Authentication (& SASL)

Basic -> username:password non cifrate

Digest -> MD5 hash, nonce

la **password** è il segreto condiviso

# La ricetta in fieri

***anchors*** dal codice OSS di mimikatz
fiumi di RAM dump/hiberfil.sys

estrazione memoria **lsass.exe**
estrazione da memoria lsass di **wdigest.dll**
ricerca *anchor*
navigazione memoria lsass
*Walk* della lista con le credenziali (*circular double linked*)

# La ricetta in fieri (II)

Volatility
HxD
Calc (!!)
Notepad++

```
C:\Users\fpi.bolt\Desktop>d:\zprj\volatility\vol.py -f UberEvil-hiberfil.sys --profile=Win7SP1x86 pslist
Volatility Foundation Volatility Framework 2.3.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x851ad020 System                    4      0     83      490 ------      0 2014-03-14 15:02:37 UTC+0000
0x86329d40 smss.exe                240      4      3       30 ------      0 2014-03-14 15:02:37 UTC+0000
0x86319d40 csrss.exe               332    324      9      448      0      0 2014-03-14 15:02:44 UTC+0000
0x86361a50 wininit.exe             372    324      3       77      0      0 2014-03-14 15:02:45 UTC+0000
0x86d63030 services.exe            480    372      9      207      0      0 2014-03-14 15:02:46 UTC+0000
0x86d66d40 lsass.exe               488    372      9      568      0      0 2014-03-14 15:02:47 UTC+0000
0x86d67488 lsm.exe                 496    372     10      141      0      0 2014-03-14 15:02:47 UTC+0000
0x86dbfbd8 svchost.exe             600    480     11      360      0      0 2014-03-14 15:02:48 UTC+0000
```

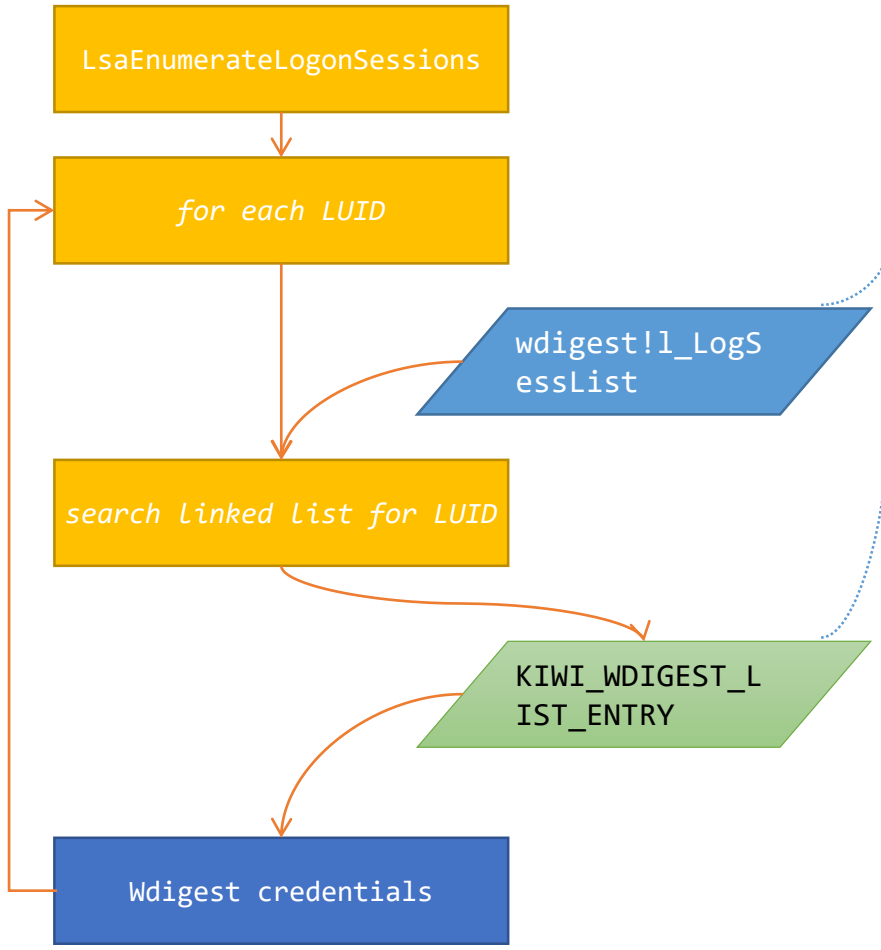
```
C:\Users\fpi.bolt\Desktop>d:\zprj\volatility\vol.py -f UberEvil-hiberfil.sys --profile=Win7SP1x86 dlllist -p 488
Volatility Foundation Volatility Framework 2.3.1
************************************************************************
lsass.exe pid:    488
Command line : C:\Windows\system32\lsass.exe
Service Pack 1

Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x00b60000     0x9000     0xffff C:\Windows\system32\lsass.exe
0x773f0000    0x13c000     0xffff C:\Windows\SYSTEM32\ntdll.dll
0x75a20000     0xd4000     0xffff C:\Windows\system32\kernel32.dll
0x754b0000     0x4b000     0xffff C:\Windows\system32\KERNELBASE.dll
0x75e70000     0xac000     0xffff C:\Windows\system32\msvcrt.dll
0x771b0000     0xa2000     0xffff C:\Windows\system32\RPCRT4.dll
0x75300000      0x7000     0xffff C:\Windows\system32\SspiSrv.dll
0x75180000    0x101000        0x13 C:\Windows\system32\lsasrv.dll
0x76da0000     0x19000        0xa2 C:\Windows\SYSTEM32\sechost.dll
0x75310000     0x1b000        0x14 C:\Windows\system32\SspiCli.dll
0x75980000     0xa0000        0x18 C:\Windows\system32\ADVAPI32.dll
0x77040000     0xc9000        0x3c C:\Windows\system32\USER32.dll
```

```
0x752e0000     0x8000        0x4 C:\Windows\system32\Secur32.dll
0x75380000     0xc000        0x6 C:\Windows\system32\cryptbase.dll
0x74ec0000    0x88000        0x2 C:\Windows\system32\kerberos.DLL
0x74ea0000    0x16000        0x5 C:\Windows\system32\CRYPTSP.dll
0x77110000    0x35000        0xb C:\Windows\system32\WS2_32.dll
0x75ca0000     0x6000       0x10 C:\Windows\system32\NSI.dll
0x74e60000     0x3c000        0x3 C:\Windows\system32\mswsock.dll
0x74e50000     0x6000        0x1 C:\Windows\System32\wship6.dll
0x74e00000    0x42000        0x5 C:\Windows\system32\msv1_0.DLL
0x74d70000     0x8c000        0x2 C:\Windows\system32\netlogon.DLL
0x74d20000     0x44000        0x3 C:\Windows\system32\DNSAPI.dll
0x74cf0000     0x22000        0x2 C:\Windows\system32\logoncli.dll
0x74cb0000     0x3f000        0x1 C:\Windows\system32\schannel.DLL
0x75560000    0x120000        0x1 C:\Windows\system32\CRYPT32.dll
0x74c80000     0x2c000        0x1 C:\Windows\system32\wdigest.DLL
0x74c40000     0x3b000        0x1 C:\Windows\system32\rsaenh.dll
```



HACKINBO

# *GentilKiwi* wdigest recipe applied




```
typedef struct _KIWI_WDIGEST_LIST_ENTRY {
       struct _KIWI_WDIGEST_LIST_ENTRY *Flink;
       struct _KIWI_WDIGEST_LIST_ENTRY *Blink;
       DWORD  UsageCount;
       struct _KIWI_WDIGEST_LIST_ENTRY *This;
       LUID   LocallyUniqueIdentifier;
       […]
       LSA_UNICODE_STRING UserName;
       LSA_UNICODE_STRING Domaine;
       LSA_UNICODE_STRING Password;
       […]
} KIWI_WDIGEST_LIST_ENTRY,
*PKIWI_WDIGEST_LIST_ENTRY;
```

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
000E5180   B0 DE 13 00 00 00 00 00 C0 12 3E FC FE 07 00 00   °Þ......À.>üþ...
000E5190   01 00 00 00 00 00 00 00 80 E1 13 00 00 00 00 00   ........€á......
000E51A0   27 5C F9 00 00 00 00 00 01 00 00 0A 02 00 00 00   '\ù.............
000E51B0   0C 00 0E 00 00 00 00 00 40 95 F0 01 00 00 00 00   ........@•ð.....
000E51C0   08 00 0A 00 00 00 00 00 20 95 F0 01 00 00 00 00   ........ •ð.....
000E51D0   10 00 10 00 00 00 00 00 E0 96 F0 01 00 00 00 00   ........à–ð.....
000E51E0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
000E51F0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
000E5200   41 00 41 00 41 00 00 00 EA 61 B9 3D A8 00 00 90   A.A.A...êa¹=¨...
000E5210   E0 74 69 01 00 00 00 00 00 BD 72 01 00 00 00 00   àti......½r.....
```



HACK IN BO

# Hai messo il sale?

Quindi in memoria le password sono in chiaro?

Nein!
Sono cifrate!

Ovviamente le chiavi sono in memoria lsass…
Da **lsasrv.dll** -> lsass.exe RAM ->
3DES Key
AES Key
IV

# Il cyberpiatto servito: mikikatz volatility

Plugin scaricabile da
http://code.google.com/p/hotoloti/
Zena Forensics
http://blog.digital-forensics.it

PoC
Windows Vista e Windows 7 (x86 e x64)
Wdigest only

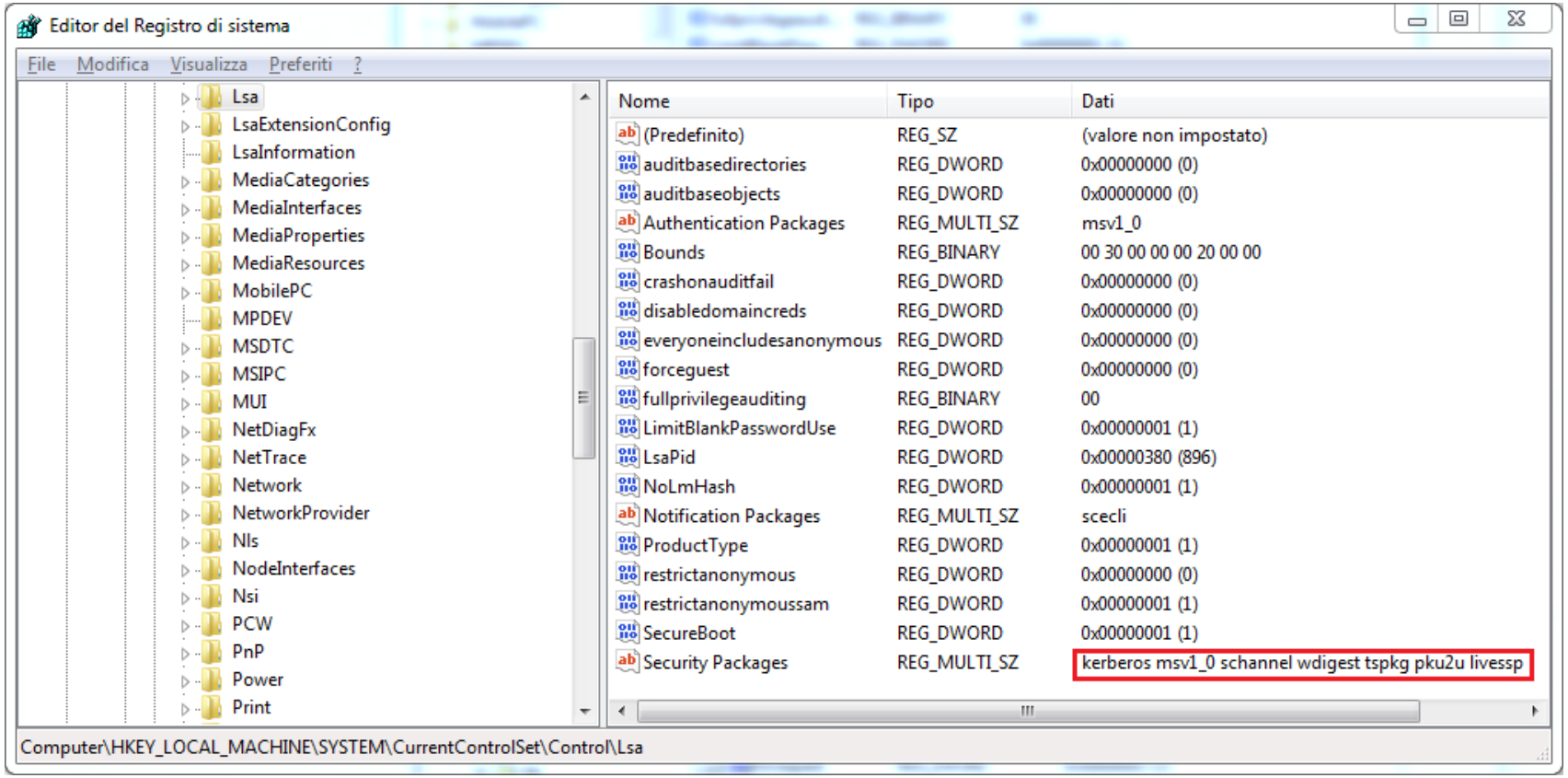
Cross platform (è python!)
Se amate Windows e WinDbg allora anche mimikatz!

# Quando mancano gli ingredienti…

- Spesso (*bias personale*) accade che le dll **lsasrv** e/o **wdigest** siano *paged-out*
- Niente più *ancore*!
- Non disperare mai… per la parte dati è meno probabile

- Con una *scansione* delle sezioni dati *heap* di **lsass** è possibile recuperare i dati utili
- Non (sempre) triviale (es: IV…)
- TODO?

# Boicottare il piatto I




**Eliminabili**

- Wdigest
- Tspkg (possibili problemi)

Dipende

- Kerberos
- livessp

Non eliminabili

- msv1_0

Da approfondire

- schannel
- pku2u



HACKINBO

# Boicottare il piatto II

«*Son due chili, che faccio lascio?*»

Il solo **msv1_0** non vi lasci tranquilli….

Da (s)vista in avanti no LM hash nel SAM
Tuttavia (svista?) LSASS ***tende*** a calcolare (e tenere in RAM)
LM hash per password inferiori a **16 caratteri**

*omo avvisato mezzo salvato*

# Ricette non ancora pronte…

Dropbox

"*A critical analysis of Dropbox software security*" F. Ledoux N. Ruff

"*Looking inside the (Drop) box*" D. Kholia

DPAPI windows7

Tool per il dump offline delle credenziali

*write once use many*

# Bis?




HACKINBO