

Computer Forensics Test di Mario Pascucci

(<http://www.ismprofessional.net/pascucci/index.php/2007/10/computer-forensic-un-piccolo-test/>)



RISOLUZIONE DI NANNI BASSETTI
<http://www.nannibassetti.com>

Non riporto tutto il testo presente sul sito di Mario Pascucci ma solo le regole d'ingaggio:

Queste le regole d'ingaggio:

1. Il pen drive è formattato con filesystem FAT
2. Non è stata usata nessuna crittografia, non ci sono password da scoprire
3. Non sono stati usati programmi sviluppati appositamente (posso assicurare che sarebbe stato impossibile recuperare alcunché dal pen drive, in questo caso)
4. Il file, se c'è, è in un formato assolutamente comune, non è stata utilizzata una applicazione dedicata agli schemi elettronici, altrimenti sarebbe stato impossibile aprire il file senza quella applicazione

Regole per aggiudicarsi la vittoria:

- verificare se vi è effettivamente uno schema elettronico nascosto nel pen-drive
- dove è nascosto
- in che formato file è memorizzato
- la procedura **ripetibile** per estrarlo dal pen drive, sempre ammettendo che il file ci sia

In questo report non sono stati usati codici hash perché visto come un gioco, ma naturalmente in un caso reale, avrei riportato tutto l'hashing dei files e dell'immagine del pendrive.

Per "sfida" personale il tutto è stato eseguito sotto Windows XP

ORA DI INIZIO DELL'INDAGINE 11:30 del 06/11/2007

Primo approccio:

Subito ho utilizzato **foremost** per un carving di base e da qui è balzato all'occhio il file *NORADI.ZIP*, file compresso e chiuso con password, ma date le regole d'ingaggio, che non prevedono file criptati o protetti, ho scartato quell'ipotesi di prova.

Poi ho utilizzato **Autopsy** (da cygwin) e montato l'immagine del pendrive (pendrive.img) come volume.

Ho cominciato a sfogliare i vari files, ad ordinarli col sorter dello Sleuthkit ed a raccogliere le stringhe per eventuali ricerche testuali.

Il sorting per file type non mi ha evidenziato molto e controllando i file types erano tutti coerenti.

Poi ho guardato i files cancellati, esaminandone il contenuto ove possibile, ma erano tutti irrecuperabili, quindi ho scartato l'ipotesi che la prova fosse tra loro.

Secondo Approccio:

Sempre con Autopsy ho fatto una ricerca per keywords sullo spazio allocato e non allocato di termini come:

elett, elettr, electr, cifr, cifrat, crypt, cript, schema, circuit, diodo, pin

Portandomi su alcuni file di testo ASCII, quasi tutti PHP, facenti parte del noto software open source WordPress.

La ricerca è stata infruttuosa tranne che per l'individuazione di un file:

geda_sym_format.h

sviluppato in linguaggio C++ che è in una cartella */olib/src* .

Analizzando i file testuali della cartella olib si possono leggere i contenuti dei file *ChangeLog* e *README* che indicano la scrittura di un programma che serve a convertire i file prodotti da **ORCAD** (noto software di progettazione elettronica) in formato **gEDA** (altro software di progettazione elettronica), ambedue i software usano file ASCII, rispettivamente con estensioni .ASC e .SYM.

Nessuna ricerca per .ASC e .SYM ha dato frutti.

Guardando il contenuto del file *geda_sym_format.h* ho ipotizzato che fosse un finto convertitore, bensì fosse un file che generava il file .sym già completo in modo da creare lo schema rubato.

L'ipotesi è decaduta poiché andava contro le regole d'ingaggio, che negavano l'uso di programmi proprietari per leggere il file ricercato.

Ancora una ricerca (Autopsy), con la keyword: "*http://*"

al fine di trovare se ci fosse un URL che rimandava ad un sito di storage, dove il ladro avesse messo l'immagine o il file dello schema elettronico.

Questa ricerca mi ha portato a trovare delle tracce di <http://> dentro dei file JPG (oltre che in tutti i file TTF ed in molti file di testo), le immagini sono:

01344_calming_1280x800.jpg e *JKHP_fedora1280.jpg*

riportavano il seguente URL:

<http://ns.adobe.com/xmp/10> sito che parla di MetaDati nascosti nelle immagini.

Ho provato un'analisi delle immagini con IRFanView e con MS Photo Info ed ho trovato solamente delle informazioni sul tipo di macchina fotografica usata, ecc.

Di nessuna utilità ai fini della mia indagine.

Terzo Approccio:

Mi viene in mente di cercare tutte le immagini con estensione **PNG** e di controllarle tutte, al fine di non avere delle finte GIF o JPG, che poi, magari, sono realmente delle PNG camuffate.

Le PNG sono un formato di immagine che permette di avere più layers vettoriali o aster, si potrebbe incollare un'immagine su un layer e poi renderlo invisibile, quindi solo aprendo l'immagine con un software di fotoritocco (**GIMP**) si può riattivare il layer nascosto e renderlo visibile.

Ma gli esami effettuati sulle PNG non hanno portato a niente.

Tra gli ultimi tentativi vado a riesaminare i file documento con estensione .DOC e trovo tre files:

diariosegretolinux.doc

principali_comandi_linux.doc

wordperf.doc (realizzato con wordperfect)

Cerco qualsiasi forma di occultamento nei files, cambio i caratteri, seleziono tutto, cerco immagini, ecc. ecc.

Poi noto che nel file: *principali_comandi_linux.doc*

Vi è un diagramma, che spiega come funziona il comando chmod di Linux, però questo diagramma sembra avere delle somiglianze con un circuito elettrico, mi sembra di intravedere la simbologia dei condensatori, ma è troppo semplice, provo a vedere se le scritte all'interno delle box, che compongono il diagramma, mi danno degli indizi, magari sostituendo qualche valore usciranno delle sigle di microchip, ma dopo varie elucubrazioni e tentativi, non mi ritrovo un'ipotesi plausibile.

A questo punto per comodità monto l'immagine del pendrive con FTK Imager della Access Data (programma freeware) per sfogliare tranquillamente i files e girando tra le cartelle noto un file particolare nella cartella **fcrackzip-03**:

dict

Un file senza estensione, che risultava essere alla analisi del tool *file* dell'ambiente CygWin e dallo Sleuthkit un formato ASCII.

Provo a farlo analizzare dal tool TridNet e mi restituisce un bel NULL, quindi non riconosce il file.

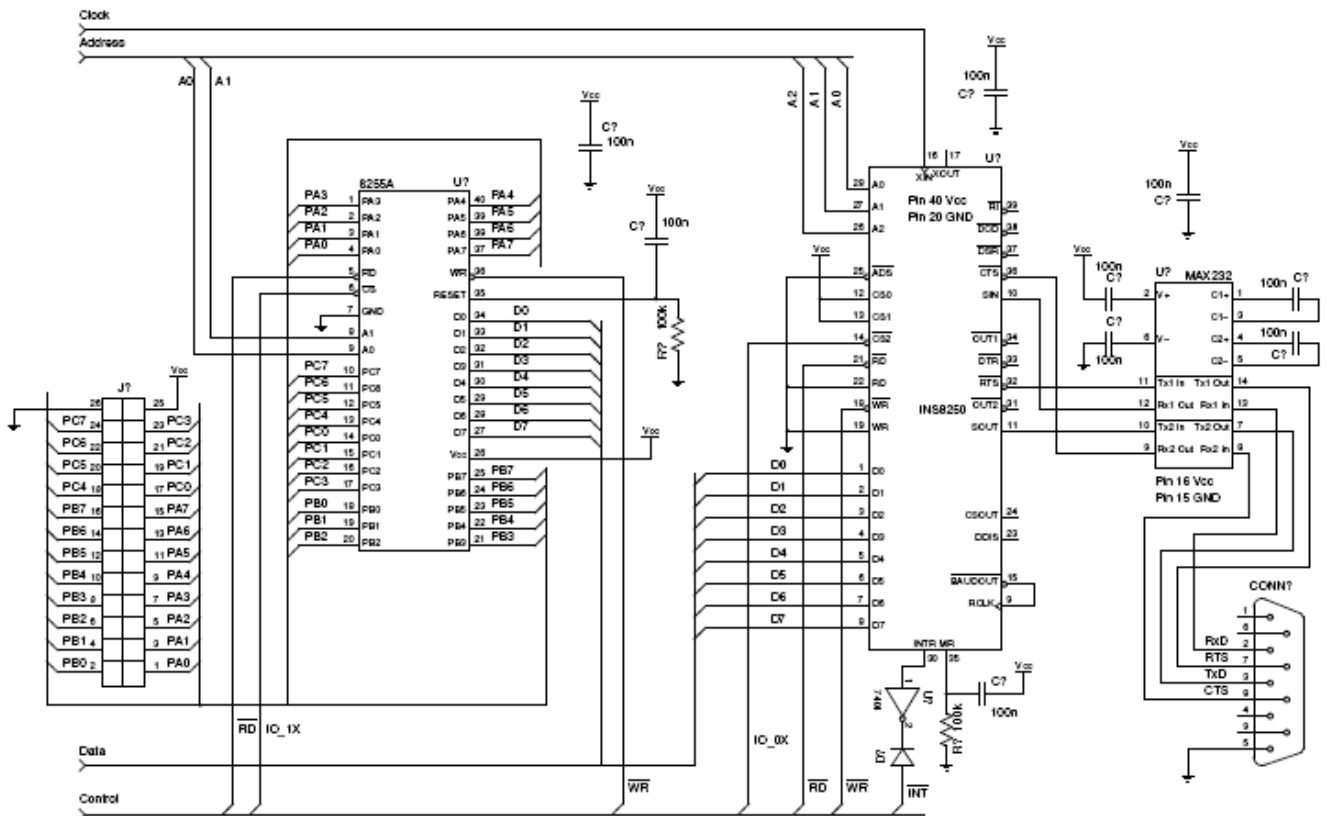
Guardo il contenuto del file e vedo l'header che inizia con **JVBER**

cercando su GOOGLE ho trovato che è un formato trasformabile in PDF tramite una codifica **base64**.

Scaricata l'utility base64 mi è bastato fare:

```
base64 -d dict > dict.pdf
```

Poi aprendo il file **dict.pdf** con Acrobat Reader è comparso lo schema elettronico.



ORA 11:26 DEL 07/11/2007 – FINE INDAGINE

Tutta l'indagine è ripetibile, i file analizzati sono sempre stati esportati dalla immagine del pendrive, quindi non è mai stata effettuata alcuna operazione in scrittura sul supporto originale.

Tools utilizzati:

FOREMOST
AUTOPSY E SLEUTHKIT
GIMP
MS PHOTO INFO
IRFANVIEW
TRIDNET
FTK IMAGER
FILE
NOTEPAD ++
BASE64
ACROBAT READER

INVESTIGATORE:

Dott. Nanni Bassetti - <http://www.nannibassetti.com> - nannib@libero.it