



Universidad Carlos III de Madrid

Reporte de Data carving - Floppy Disk

Juan Diego Llano Miraval

Fecha: 18/05/2024

Procedure

The first step was checking the sha256:

```
root@z:/home/zud/Desktop/floppy# sha256sum floppy.dd
15fab9d7c0993b02ca720b7e6d014b584874553fbacf6bca457f8b85fa89a940 floppy.dd
```

it is correct to say this is the same as the provided, so we mount the disk into autopsy, and got 3:

Listing	
img_floppy.dd/\$CarvedFiles/1	
Table	Thumbnail Summary
Name	
✖ f0000033_JOHN_by_the_grace_of_God_King_of_Eng	
✖ f0000165_Four_score_and_seven_years_ago_our_fatl	
✖ f0000207.xls	

I also found another document but it was a copy of the second document in the image. From this we can see 2 documents of text, and 1 microsoft excel file. When we checked the first file:

it is just text that is not relevant to the case, but in the metadata we can confirm that Emma Crook was the creator of the file, so she might be the owner of the floppy disk. The second file is similar as the first one, there is no relevant information related to the investigation:

-----METADATA-----

2 / 5

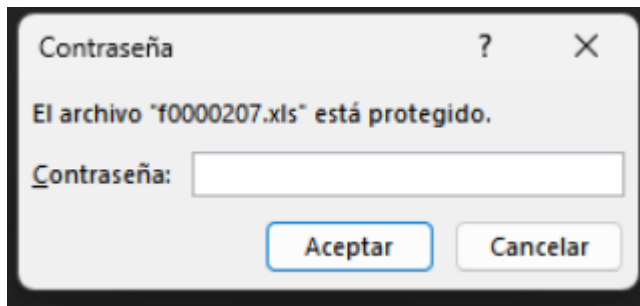
"I @\$
\pPz

vZzj
qfE[
#_lx
]WZQ
1hq*
TC}=
Y:N9
2'qY`
h\$(O
>+4c
>n.v
qwzE
T_04
q;)S
""t
^zUk
~3E^
.z'7u
>`k;
%e4
nPM%d
nr3+
jNz)
EEe}
ij=D'
L6i<&
WWc=7
9t9^
t}*
<qqa
l".)
7PL[g\$
DVJ?
-xTH
Emma Crook
Microsoft Excel@
Really Big Companyh
Sheet1
Sheet2
Sheet3
Worksheets
_PID_GUID
{899F2B7A-0723-11D9-9157-00045A8C9A41}
Root Entry
Workbook
SummaryInformation
DocumentSummaryInformation

we exported the .xls for further analysis. We extracted the sha256 from the excel file before proceeding:

```
(root@kali)-[/home/kali/Desktop]
# sha256sum f0000207.xls
56e1b34930c4742636fd0c327ac0f9926c7acf00ab5fb6723a7a7227ff74c8da  f0000207.xls
```

We confirmed that this is a microsoft excel file and we proceed to open. When we open it requests a password, so it is encrypted:



We took out the hash of the file with the help of john the ripper:

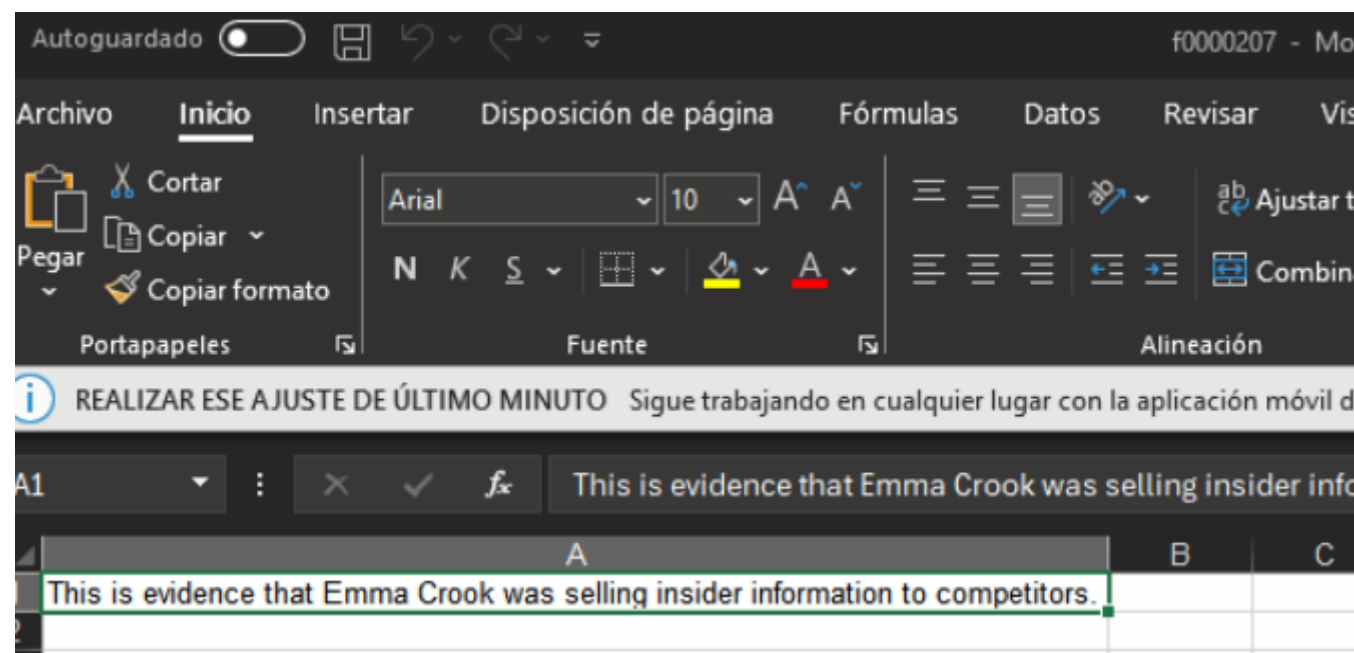
```
(root@kali)-[/home/kali/Desktop]
# python /usr/share/john/office2john.py f0000207.xls
f0000207.xls:$oldoffice$0*fd8366d381a091ec761dacf4501b79ca*4ab94798dc6cca226c
204024a2829c74*46854c3e0b40d8c236b547756a89e8ad::::f0000207.xls

(root@kali)-[/home/kali/Desktop]
# python /usr/share/john/office2john.py f0000207.xls > hash.txt
```

And by running a simple analysis (by accident i was checking the hash was correct) john the ripper got the password from a default password list:

```
(root@kali)-[/home/kali/Desktop]
# john hash.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (oldoffice, MS Office ≤ 2003 [MD5/SHA1 RC4 32/64])
Cost 1 (hash type [0-1:MD5+RC4-40 3:SHA1+RC4-40 4:SHA1+RC4-128 5:SHA1+RC4-56]
) is 0 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:ASCII
crook (f0000207.xls)
1g 0:00:00:00 DONE 3/3 (2024-05-18 07:00) 1.265g/s 325278p/s 325278c/s 325278
C/s jamum..cez11
Use the "--show --format=oldoffice" options to display all of the cracked pas
swords reliably
Session completed.
```

Another strategy I was planning to use, was an incremental brute force attack. We have finally opened the excel file and got:



This is the evidence requested that Emma Crook was selling information.