

Sticky-situation

sticky-situation

991

forensics hard

- Somebody found this laptop running in the basement of Cullimore - right by the mailroom.
- Normally we'd just turn it in, but with all of the cyberattacks and cyberpsychosis going on - we're not about to throw away evidence without examining it first.
- Examine this ad1, and see if anything sticks out to you.
- I find it useful to write down my thoughts and findings on whatever I have nearby - paper, sticky notes, etc.
- Once I have evidence, I put it on a wall or lay it on the table and start connecting the dots.
- Yes, I'm crazy with strings all over the wall - but it helps!
- Maybe you'll find my approach useful, too?

File download: [sticky-situation.ad1](#)

Flag Format: `jctf{exactly_as_it_appears}`

Pretty cool challs, given ad1 file we are tasked to find what the user store in sticky notes in his device. So we are gonna find out the artifact of the sticky notes.

First of all im just gonna mount it using FTK Imager and analyze it using ArtiFast Lite.

From this article (<https://dingtoffee.medium.com/windows-sticky-notes-forensics-80ee31ab67ef>)

we know that there are some artifacts related to Sticky notes located

at%APPDATA%\Roaming\StickyNotes\and

%LOCALAPPDATA%\Packages\Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe

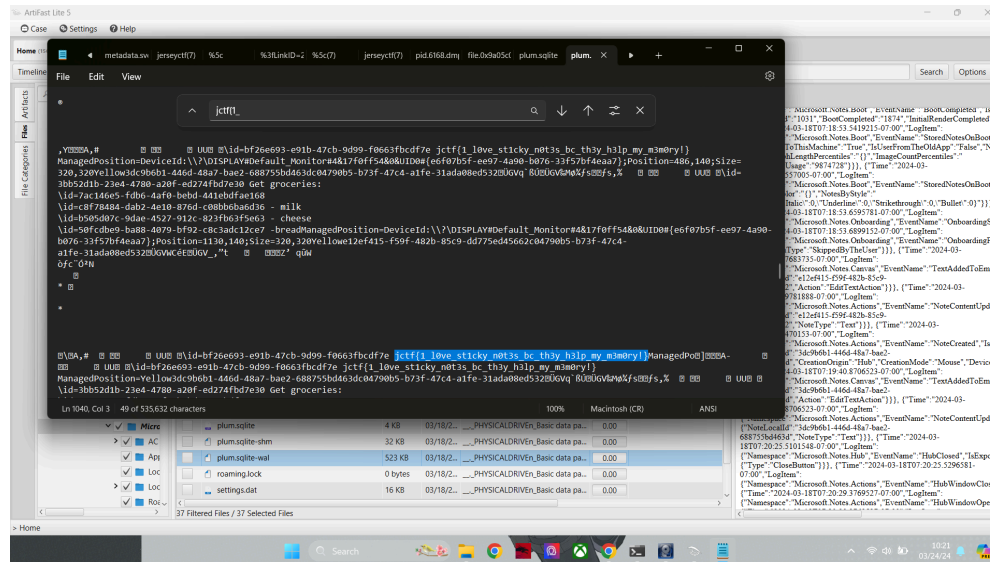
We will check both of the directory

Transaction Log and Events :

```
%LOCALAPPDATA%\Packages\Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe\LocalS  
tate\plum.sqlite-shm
```

```
%LOCALAPPDATA%\Packages\Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe\LocalS  
tate\plum.sqlite-wal
```

Because of we are searching for what the user stored in the sticky notes, we will check at the "plum.sqlite-wal" file and open it using notepad



All along watchtower

All-Along-the-Watchtower

561

forensics medium

Our informant has met back with us at base. He has stolen a file from a suspect who's starting to show dangerous signs of cyberpsychosis. Aside from his unending lust for bloodshed, our informant noted that the suspect would not shut up about "rats making him crazy" as well as the term "Base FFFF+1" but that's probably just nonsense. Anyways, can you look beyond the layers of this corrupted file to extract the information that we need?

Developed by: [BonsaiUmai](#)

Given the file we're just gonna brute it using given password.txt and john

```
jons@01-20-jonathans: ~/jers x jons@01-20-jonathans: ~/vol3, x + v
--max-length=N Request a maximum candidate length in bytes
--max-candidates=[-]N Gracefully exit after this many candidates tried.
--max-run-time=[-]N (if negative, reset count on each crack)
--mkpc=N Gracefully exit after this many seconds (if negative,
--no-loader-dupecheck Request a lower max. keys per crypt reset timer on each crack)
--pot=NAME Disable the dupe checking when loading hashes
--regen-lost-salts=N Pot file to use
--reject-printable Brute force unknown salts (see doc/OPTIONS)
--tune=HOW Reject printable binaries
--subformat=FORMAT Tuning options (auto/report/N)
--format=[NAME][CLASS][...] Pick a benchmark format for --format=crypt
Force hash of type NAME. The supported formats can
be seen with --list=formats and --list=subformats.
See also doc/OPTIONS for more advanced selection of
format(s), including using classes and wildcards.

(jons@01-20-jonathans) [~/jersey/_the-panglao-watchtower.jpg-0.extracted/SECRETS]
$ john hash.txt --wordlist=password.txt
Using default input encoding: UTF-8
Loaded 1 password hash (7z, 7-Zip archive encryption [SHA256 512/512 AVX512BW 16x AES])
Cost 1 (iteration count) is 524288 for all loaded hashes
Cost 2 (padding size) is 5 for all loaded hashes
Cost 3 (compression type) is 2 for all loaded hashes
Cost 4 (data length) is 683 for all loaded hashes
Will run 12 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
jimihendrix123! (protected_2.7z)
1g 0:00:00:01 DONE (2024-03-24 15:11) 0.6666g/s 128.0p/s 128.0c/s 128.0C/s jimihendrix..Jimihendrix07
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Extracting it using the password give us file “secrets.txt” contains some weird strings. Based on the chall description “Base FFFF+1” refers to Base65536 so we gonna decode it using that

[illegible]