Since I (a) never trust any browser or site and (b) the article indicated that there is a $5 fee to avoid the “single random transaction removal”, I felt compelled to throw together an R script to do this for at least folks who are capable of setting up R so that all processing is guaranteed to be local.

FWIW the site does appear to do what it says on the tin (all processing is, indeed, local). That doesn’t mean one of your extensions isn’t spying on you, nor does it mean that the site could not turn evil someday (one its own or via an attacker compromise).

```
read_apple_card_statement <- function(path) {

  require(stringi)
  require(pdftools)
  require(tidyverse)

  # make sure the file exists
  path <- path.expand(path[1])
  if (!file.exists(path)) stop("File '", path, "' not found.", call.=FALSE)

  pdf_text(path) %>% # read it in
    stri_split_lines() %>% # turn \n to a separate character vector element
    unlist() %>% # flatten it
    stri_trim_both() %>% # get rid of leading/trailing spaces
    keep(stri_detect_regex, "^([[:digit:]]{2}/[[:digit:]]{2}/[[:digit:]]{4})")
%>% # find lines that start with a date
    map_df(~{
      rec <- as.list(unlist(stri_split_regex(.x, "[[:space:]]{3,}"))) # find the
columns
      if (stri_detect_fixed(.x, "%")) { # lines with a `%` in them distinguish
charges from payments
        rec <- set_names(rec, c("date", "description", "daily_cash_pct",
"daily_cash_amt", "amt")) # ones with charges have cash back columns
      } else {
        rec <- set_names(rec, c("date", "description", "amt")) # ones w/o % do
not
      }
    }) %>%
    mutate(
      date = lubridate::mdy(date), # make dates dates
      amt = stri_replace_first_fixed(amt, "$", "") %>% parse_number(), # dollars
to numbers
      daily_cash_pct = parse_number(daily_cash_pct)/100, # % to numbers
      daily_cash_amt = parse_number(daily_cash_amt) # dollars to numbners
    )

}

list.files("~/Downloads", pattern = "Apple Card Statement", full.names = TRUE)
%>%
  map_df(read_apple_card_statement)
```

You can send the PDF statements from the Apple Card app to your Mac via Air Drop and it will put them into `~/Downloads`. I recommend putting them somewhere else since you’ve likely given all sorts of applications access to `~/Downloads` when prompted to on Catalina (yay security theatre). Wherever you put them, you can read them individually with `read_apple_card_statment()` or you can then just `list_files()` and bind all the individual statements together:

```
list.files("~/WhereYouPutAppleCardStatements", pattern = "Apple Card Statement",
full.names = TRUE) %>%
  map_df(read_apple_card_statement)
```

FIN

Be very wary of what you put your trust into online. Just because a site is benign one day does not mean it won’t be malicious (deliberately or otherwise) the next. Also, lobby Apple to provide data in more useful formats, especially since it provides applications like Numbers for free with their operating system.