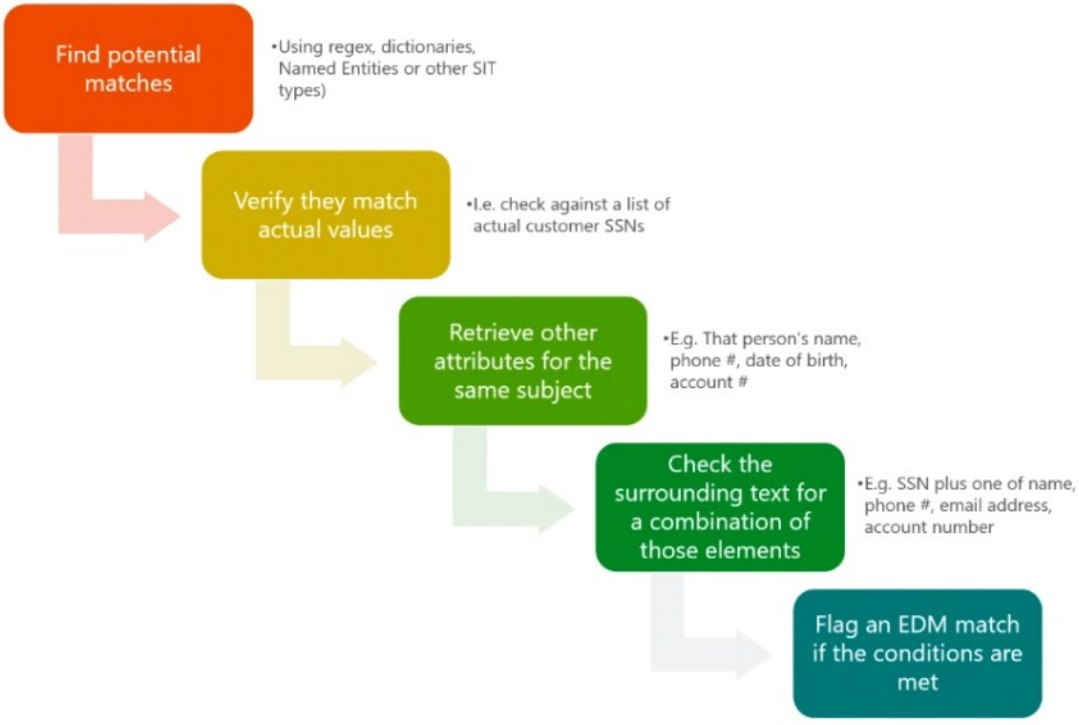
EDM part 2

mercredi 1 juin 2022 17:06

EDM at work



General best practices

Creating your data table:

#1: Use TSV file for your sensitive file, not CSV

- If records have commas in values (e.g. street address) or single or double quotes (e.g. O'Connor, John "Hannibal" Smith, etc.) CSV is tricky to get right. TSV rarely causes problems even in these cases.
- Surround all columns within double quotes just for redundancy, EDM will strip them out before hashing.

#2: Sanitize your data *automatically*

- Identify issues with your table such as corrupt data, fields that need to be divided or surrounded in quotes, etc., and take note of the issues.
- Either convince the data owners to fix it up at the source, or build a script to fix the data in bulk, you don't want to be doing it manually on a weekly basis.

Hashing and importing your data table:

#3: Install the EDM upload agent in a custom folder

- If you install it in the default folder you will need admin privileges to run it, since it is under Program Files and the tool writes logs in the same folder.

#4: Validate your data before hashing to detect potential format issues (e.g. missing or extra columns)

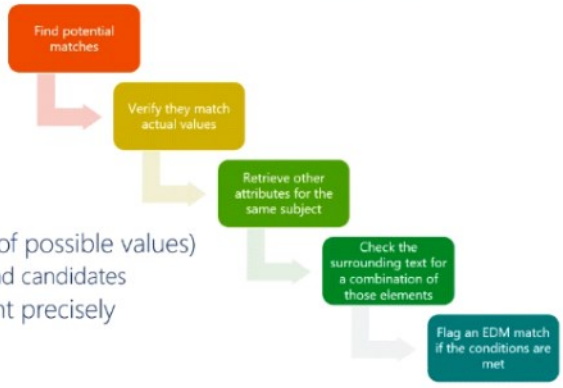
- `EdmUploadAgent.exe /ValidateData /DataFile [data file] /Schema [schema file]`

#5: Separate hash and upload processes

- The EDM table is extremely sensitive before hashing, do not put it in an internet-facing computer
- Hash in an internal machine, delete the original file, copy the hashed file to an internet facing computer and upload from there.
- Build a script that does it for you every time!

The challenge with the primary element

- Each SIT pattern has a primary evidence field
- Match candidates will be matched using a SIT first



- That field must meet *all* the following conditions:
  - Marked in the schema as searchable (so it's indexed)
  - The values must be "relatively" unique (e.g. have many thousands of possible values)
    - Date of birth, gender, marital status, first name, nationality, e.g. are all bad candidates
  - The values must be detectable using a SIT that matches the content precisely
    - Must detect all values
    - Not too many false positives (e.g. "four digit number")
    - Not too common in content (e.g. no more than 100 documents/emails per second with matches on average)
  - The values in the table must match what's in content as-is in the documents (e.g. if Full name is listed as Firstname Lastname, it will not match "Lastname, Firstname").

Key considerations for the "trigger" SIT

- The SIT used for a primary element must not be too frequently present in content
  - E.g. not in every document or email generated.
  - Keep in mind that document metadata and text in email headers are also included in matching!
    - Every email and document has multiple dates, email addresses, GUIDs, IP addresses, names, etc.. Make sure you are not using a SIT that will detect those!
      - SIT might also be improperly firing because it is finding "substrings", e.g. \d{6} will detect six consecutive digits within another string, like a GUID, \b\d{6}\b will only detect six digits alone.
- The SIT must match the whole string as present in the table and nothing else
  - Or else, it will produce a hash that's different from what is stored.
  - E.g. [a-z]+\@[a-z]+\.[a-z]+ will detect only the highlighted part in the email address: john.smith@company.co.uk
  - A regex starting and/or ending with \s will include the space character as part of the match! Use \b instead.
- The SIT must match with and without any optional delimiters
  - The "ignored delimiters" option only strips the characters after a candidate is detected
    - E.g. if Ignored delimiters is set to "-", the string 123-45-6789 will be stripped of the dash before hashing, but if the SIT is looking for \d{9} it won't even flag it as a candidate.