Microsoft

Accurate data classification using Exact Data Matching

Enrique Saggese
Principal Program Manager, CxE

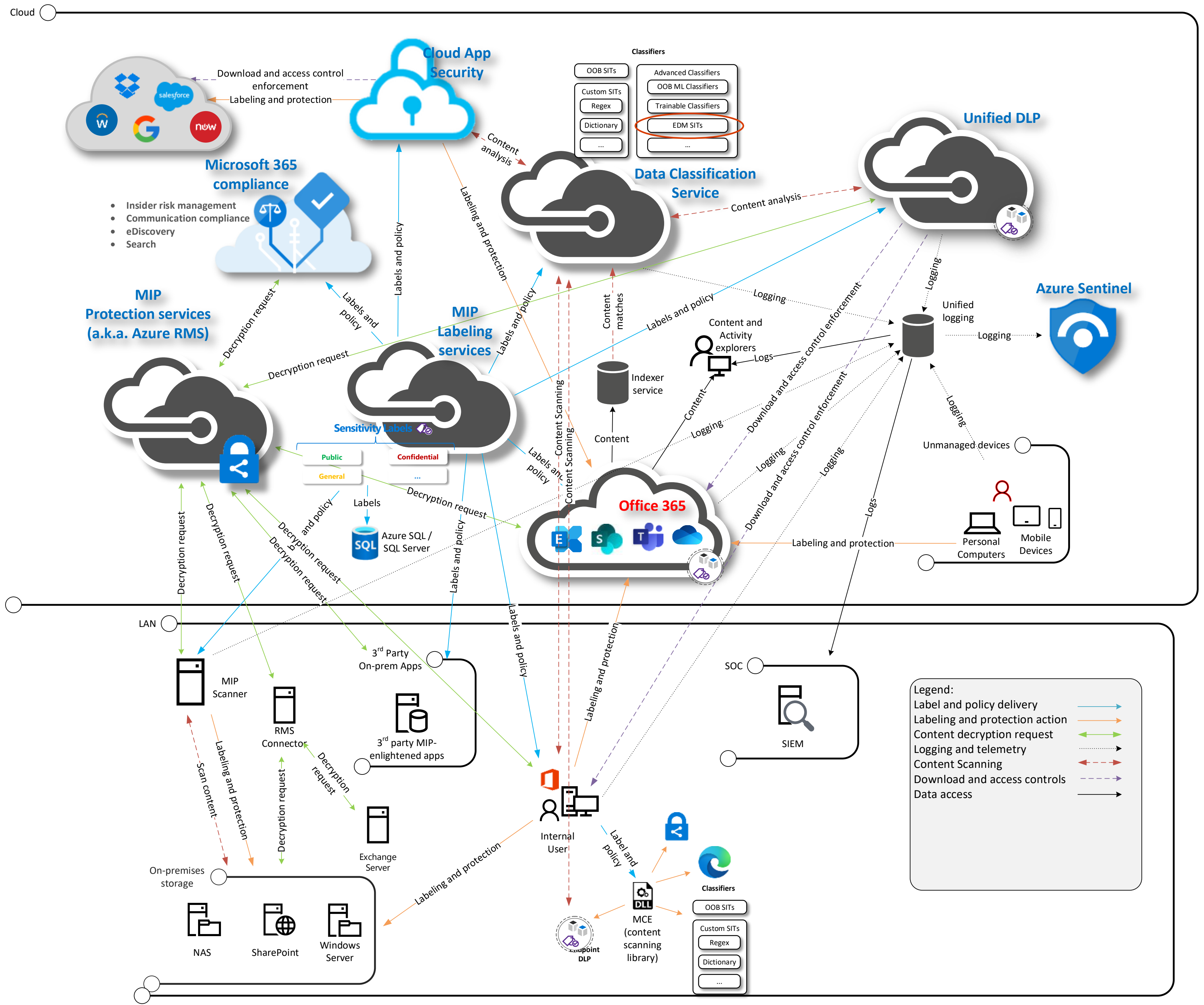
Microsoft Purview
Information Protection and Compliance
Microsoft Corp.

Agenda

- What is EDM
- EDM under the hood
- How to configure EDM
- EDM in specific industries
- Tips, tricks and advanced techniques for deploying EDM
- Sneak peek: the new EDM wizard

What is exact data matching

Data classification in the Microsoft Purview architecture



Sensitive information classifiers in Microsoft Purview

- Detection of sensitive content can be done using a multitude of mechanisms:
  - Built-in sensitive information types in Microsoft 365 (200+)
  - Custom sensitive information types
    - Using regular expressions
    - Using predefined functions
    - Keyword dictionaries
    - Keyword lists
    - Combinations of those (e.g. main criteria plus additional evidence requirements) and built-in validators
  - Trainable classifiers
  - Pre-trained ML classifiers (40+)
  - Named entity recognition
  - Exact data matching
  - Form fingerprinting (for email attachments)
  - More coming (e.g. Advanced fingerprinting, OCR)
- Humans are also good at detecting sensitive content
  - Manually applied labels are also a form of classification
  - But humans are lazy

False positives vs. false negatives

- A false positive is when content matches a rule you would not expect to match.
- Two kinds of false positives:
  - Classifier match false positive: the SIT (or other classifier) detects content that is not what you expect.
    - E.g. your company has a part # in the form nnn-nn-nnnn in a product that includes the letters “SSN” in its name, which causes such part numbers to be flagged as US Social Security Numbers even though they aren’t.
  - Functional false positive: The SIT detected pieces of information that are of the desired type, but not relevant.
    - E.g. you have a DLP rule to block sharing of PII, and it fires when someone shares their own SSN with their tax advisor.
- A false negative is when you miss detecting something you should have

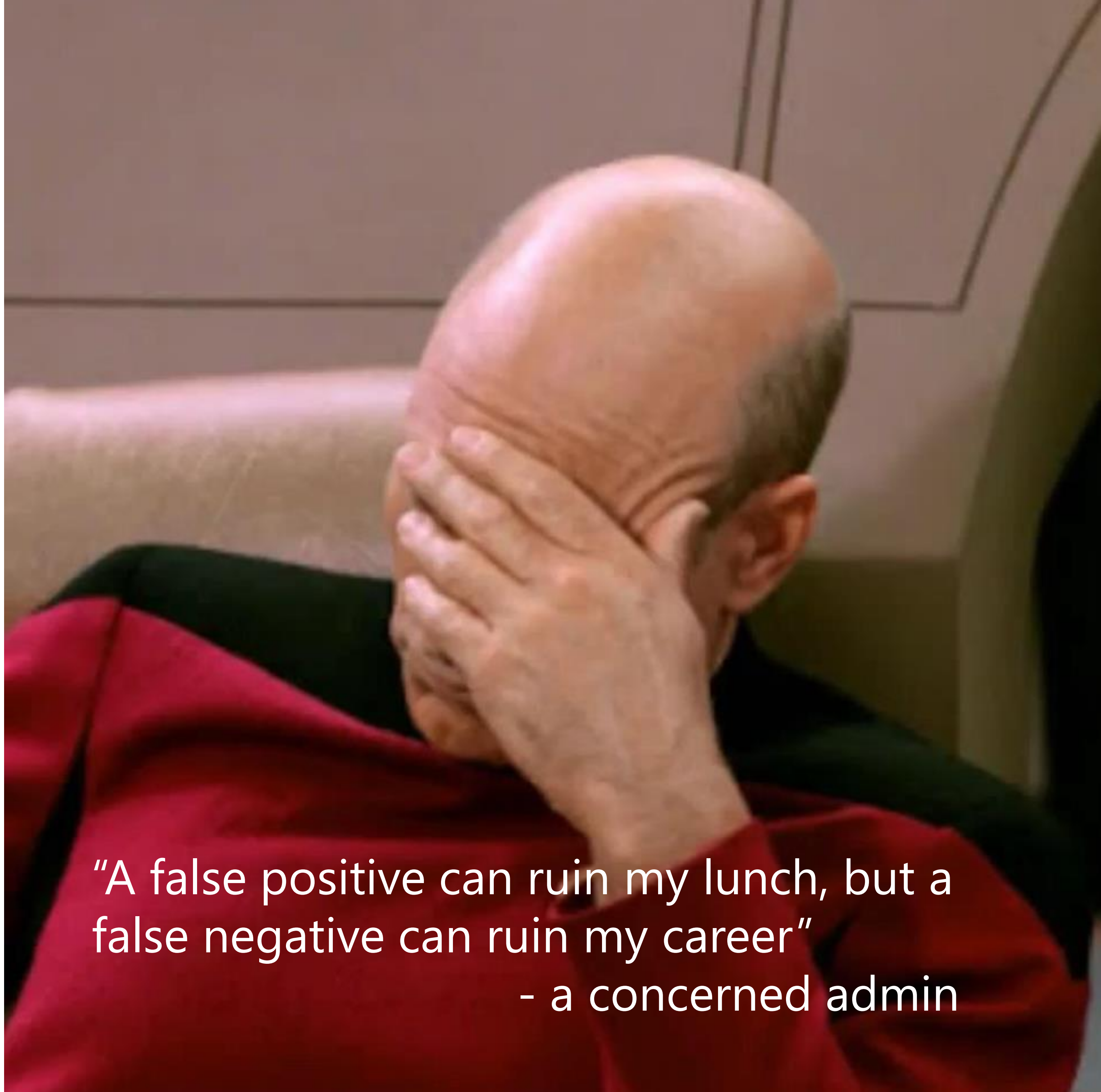
The costs of false positives

Business disruption: users are prevented from performing legitimate actions

IT overload: too many alerts take too much time to review and assess.

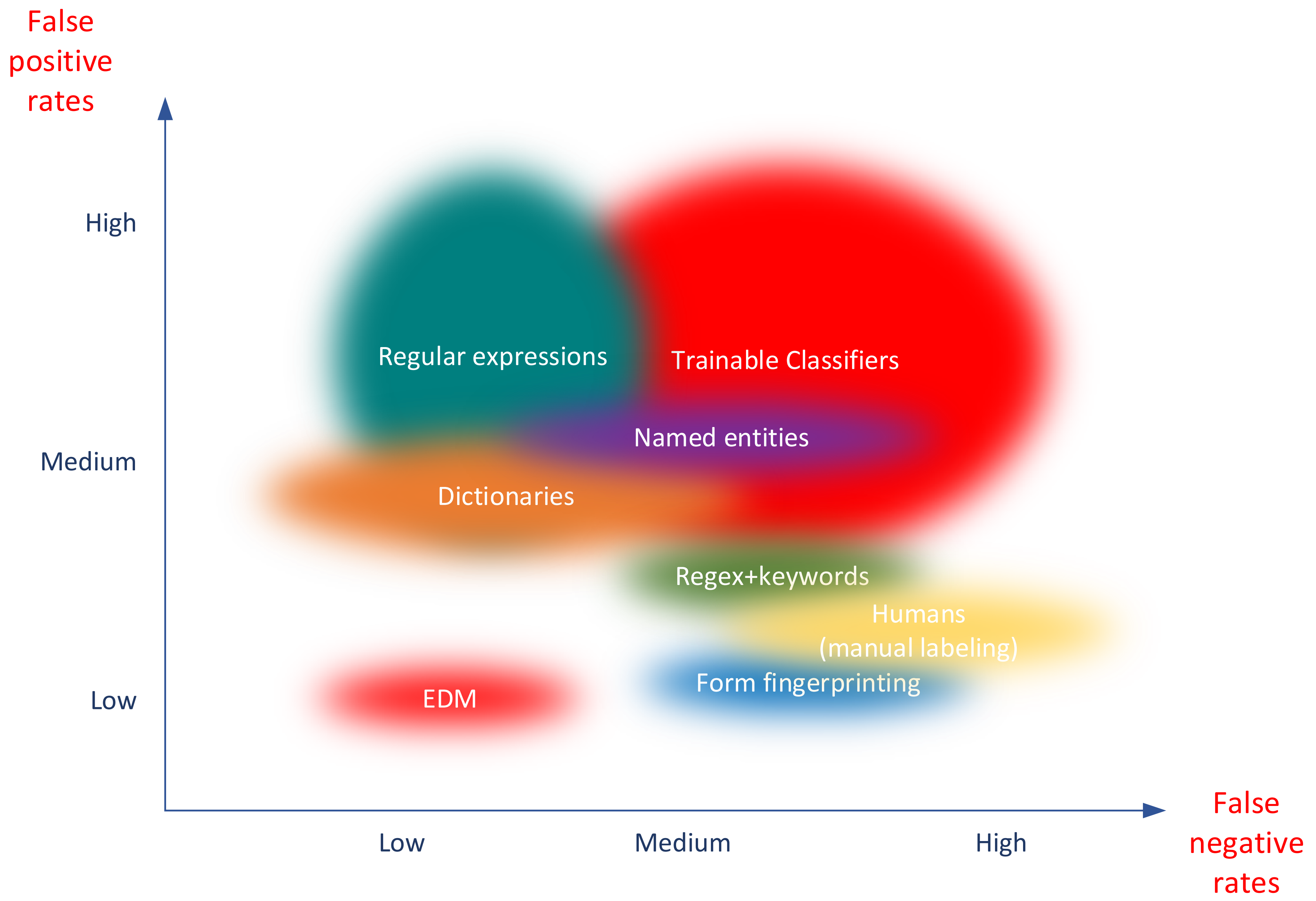
Missing the needle in the haystack: too many matches or too many alerts make you dismiss the important ones.

Increase in false negatives: to offset FPs you make your rules less sensitive, which results in false negatives.



"A false positive can ruin my lunch, but a false negative can ruin my career"
- a concerned admin

Classifier vs classifier: how they compare



Note: there are other fundamental differences between classifier types, so this is not an apples to apples comparison

So EDM is better, right?

Sometimes. EDM is useful for identification and protection of sensitive info about *known subjects*, e.g.:

- Customer data (PII)
- Employee data (PII, PHI, employment info, performance data)
- Patient data (PHI)
- Device info (e.g. subscriber device, servers and equipment)
- Customer affinity program data
- Population PII (in government organizations)
- Customer account data (e.g. account IDs)

And many more

EDM can't be used for "general" PII (e.g. not your customers). You need to have a source for what you want to detect.

Markets with highest adoption:

- Health care providers
- Health care payors
- Insurance
- Financial services
- Retail
- Hospitality and travel
- Consumer services
- HR in a variety of markets
- Professional services

Where you can use EDM

- Data loss Prevention
  - Exchange, SharePoint, OneDrive, Teams chat and Endpoint DLP policies.
  - Microsoft Cloud App Security DLP - for third party cloud apps.
- Auto labeling
  - Auto apply a sensitivity label in SharePoint and OneDrive data at rest
  - Auto apply a sensitivity label in Exchange Online to data in transit
  - Client-side autolabeling in Office apps
- Data discovery in Content Explorer
- Coming soon:
  - Advanced eDiscovery
  - Insider risk management

How EDM works

Requirements for EDM

- A table with one or more columns of data for each subject
  - Data must be “clean” (i.e. more or less consistently formatted and complete)
  - You must be able to export the data to a comma, tab or pipe separated text file
  - You *do not* need to supply that data to Microsoft or upload it to your tenant
- Run some tools on the data
  - Hash and upload process explained later
- Privileges
  - Must have tenant or compliance admin privileges to configure EDM
  - Updating the data doesn’t require the data, can be controlled via special group
- Licensing
  - Microsoft 365 E5
  - Microsoft 365 Information Protection and Compliance
  - Office 365 Advanced Compliance

EDM at work

- Find potential matches
  - Using regex, dictionaries, Named Entities or other SIT types)
- Verify they match actual values
  - I.e. check against a list of actual customer SSNs
- Retrieve other attributes for the same subject
  - E.g. That person’s name, phone #, date of birth, account #
- Check the surrounding text for a combination of those elements
  - E.g. SSN plus one of name, phone #, email address, account number
- Flag an EDM match if the conditions are met

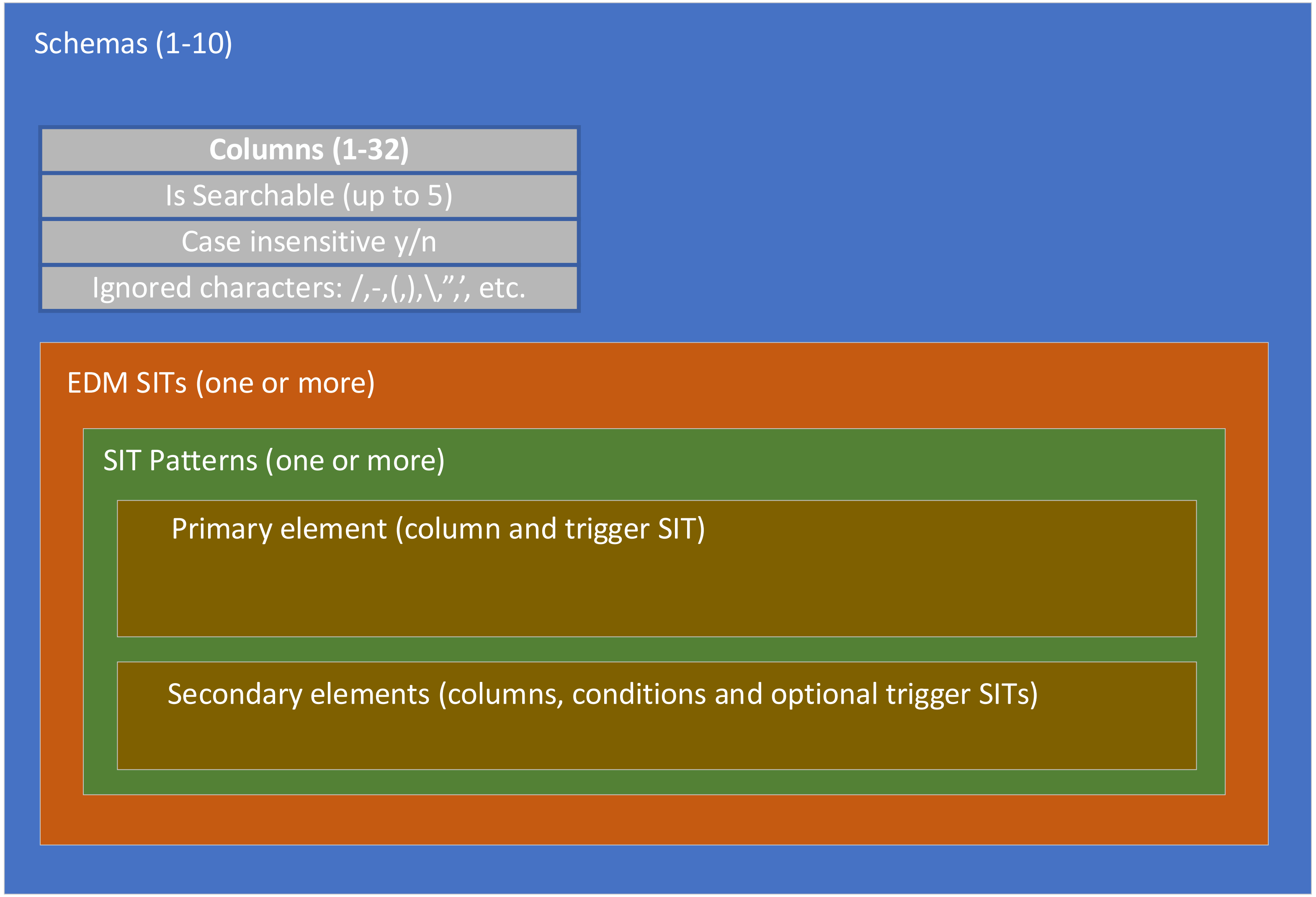
What’s in an EDM configuration

- EDM Schemas (up to 10 per tenant)
  - Definition of what columns compose the sensitive data and their propeties
  - Up to 32 columns, up to 5 searchable
- EDM datastores (one per schema)
  - A table of *hashes* of sensitive data to use for lookups
  - 100M rows max (not enforced, actual limit is 500M total cells)
- EDM SITs (no limit, can be multiple per schema)
  - One or more “patterns” per SIT
  - Based on a regular SIT, but with a lookup on a column in the datastore to refine matches.
  - Can include a single condition (column) or also additional evidence (content matching multiple columns for the same row).

Detour: why do we have to identify “potential” matches?

- Can’t we just check everything for an exact match?
  - We could, but it would be computationally impractical and slow.
  - A match can be a word or number, multiple words, part of a word, etc.. Each document contains $(n^2+n)/2$ sequences of strings inside (not counting ignored delimiters or casing)
  - If your company has 50,000 employees producing 100 pieces of content (email or document) per second, each with 500 words, that is equivalent to three quintillion strings to check per day.
  - If your table has 100 million rows and ten columns to check against each… you get the idea.
- But can’t you optimize it? Not all sequences make sense!
  - That’s what we did: you tell us (via a SIT) what’s a meaningful string to check.

Anatomy of an EDM SIT


Schemas (1-10)
Columns (1-32)
Is Searchable (up to 5)
Case insensitive y/n
Ignored characters: /,-,(,),\,”,’, etc.
EDM SITs (one or more)
SIT Patterns (one or more)
Primary element (column and trigger SIT)
Secondary elements (columns, conditions and optional trigger SITs)

Anatomy of an EDM SIT: example

EDM Datastore: Customer PII
Schema:

	SSN	Lastname	DoB	Address	Accountnumber
Searchable	X				X
Case insensitive		X		X	X
Ignored characters	-		/ and -		/ and -

EDM SIT: Customer biographical data

Pattern 1: SSN and DoB and/or Last Name

Primary Element: SSN
Based on: US Social Security Number SIT
Match against: SSN column (searchable column in schema)

Secondary elements: Lastname, DoB
Requirements: one of two

Pattern 2: SSN and street address

Primary Element: SSN
Based on: US Social Security Number SIT
Match against: SSN (searchable column in schema)

Secondary elements: Address

EDM SIT: Customer account data

Pattern 1: Account number and name

Primary Element: Accountnumber
Based on: Custom SIT (Account Nr., regex “\b[ABCD]\-?\d{6}\b”)
Match against: Accountnumber (searchable column in schema)

Secondary elements: Lastname

Pattern 2: Account number and SSN

Primary Element: Accountnumber
Based on: Custom SIT (Account Nr., regex “\b[ABCD]\-?\d{6}\b”)
Match against: Accountnumber (searchable column in schema)

Secondary elements: SSN

About the sensitive info table

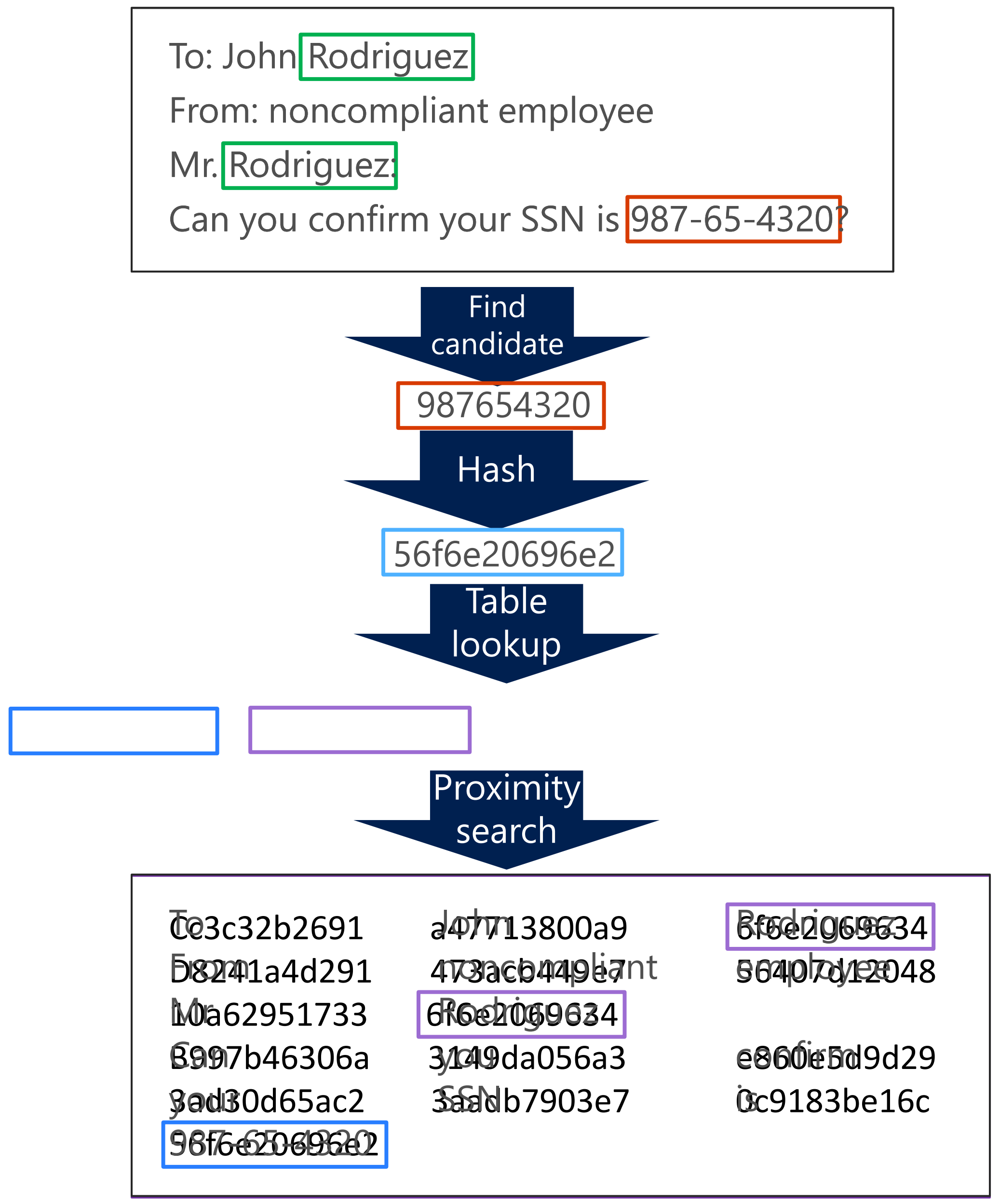
- For EDM to work we *must* be able to check candidate matches against your data.
- But we don’t need your data!!!!
- We can use this little trick called hashing
  - A hash is a non-reversible but quasi unique transformation of a string:
  - E.g.:
  - The cat in the hat => 7a652063617374e0696e207468f52f68617
  - The cat in the hut => bb2206c6d20cd04bb46t6161ae206c21db
  - 7a65206361737420696e207468652068617 ≠> The cat in the hat
- No other plausible text matches those hashes
- Only way to find out the original value is to hash all possible values and compare
- A hash can be “salted” by adding a fixed value to each string before hashing, to make the transformation unique to the customer

How hashing is used in EDM

SSN	LastName	DoB	Account#
987-65-4320	Rodriguez	2/14/1980	ABC19838372
078-05-1120	Smith	3/22/1957	BAD38229209
219-09-9999	Zhang	11/10/2001	AAB39383894

Hash and Upload

SSN	LastName	DoB	Account#
56f6e20696e2	6f6e2069634	0636f6d70617	2e4e45542c20
4b177e0db1d	82517c9053d	9e7280c96dd	39750ed0c4e
ebc4103dda7	6e206f662073	580e00857dd	3893eb087db

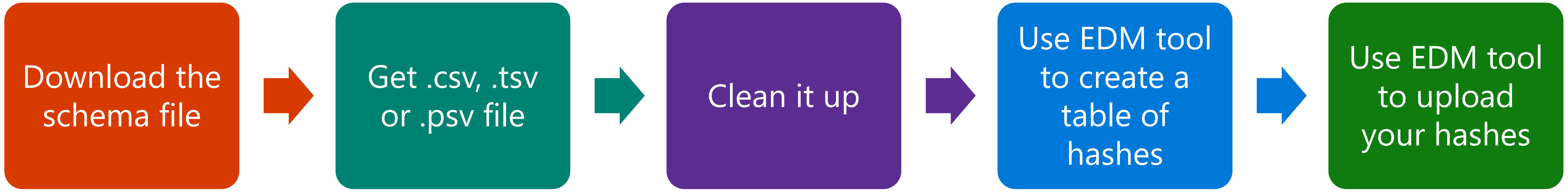


To: John Rodriguez
From: noncompliant employee
Mr. Rodriguez:
Can you confirm your SSN is 987-65-4320?

Deploying EDM

Deploying EDM (short version)

- Step 1: Create your schema
  - From compliance center or via PowerShell
- Step 2: Define your sensitive info types
  - Can be done after step 3 or more likely in parallel
- Step 3: Hash and upload your data


Download the schema file
Get .csv, .tsv or .psv file
Clean it up
Use EDM tool to create a table of hashes
Use EDM tool to upload your hashes

- Step 4: Profit!

Define EDM Schema (the hard way)

```xml
<?xml version="1.0" encoding="utf-8"?>
<EdmSchema xmlns="http://schemas.microsoft.com/office/2018/edm">
  <DataStore name="SampleDataStore" description="Sample Datastore" version="1">
    <Field name="id" unique="true" searchable="true" />
    <Field name="firstname" unique="false" searchable="true" />
    <Field name="lastname" unique="false" searchable="false" />
    <Field name="title" unique="false" searchable="false" />
    <Field name="dob" unique="false" searchable="false" />
    <Field name="creditcard" unique="false" searchable="true" />
    <Field name="ssn" unique="false" searchable="true" />
  </DataStore>
</EdmSchema>
```

Sample EDM Schema xml

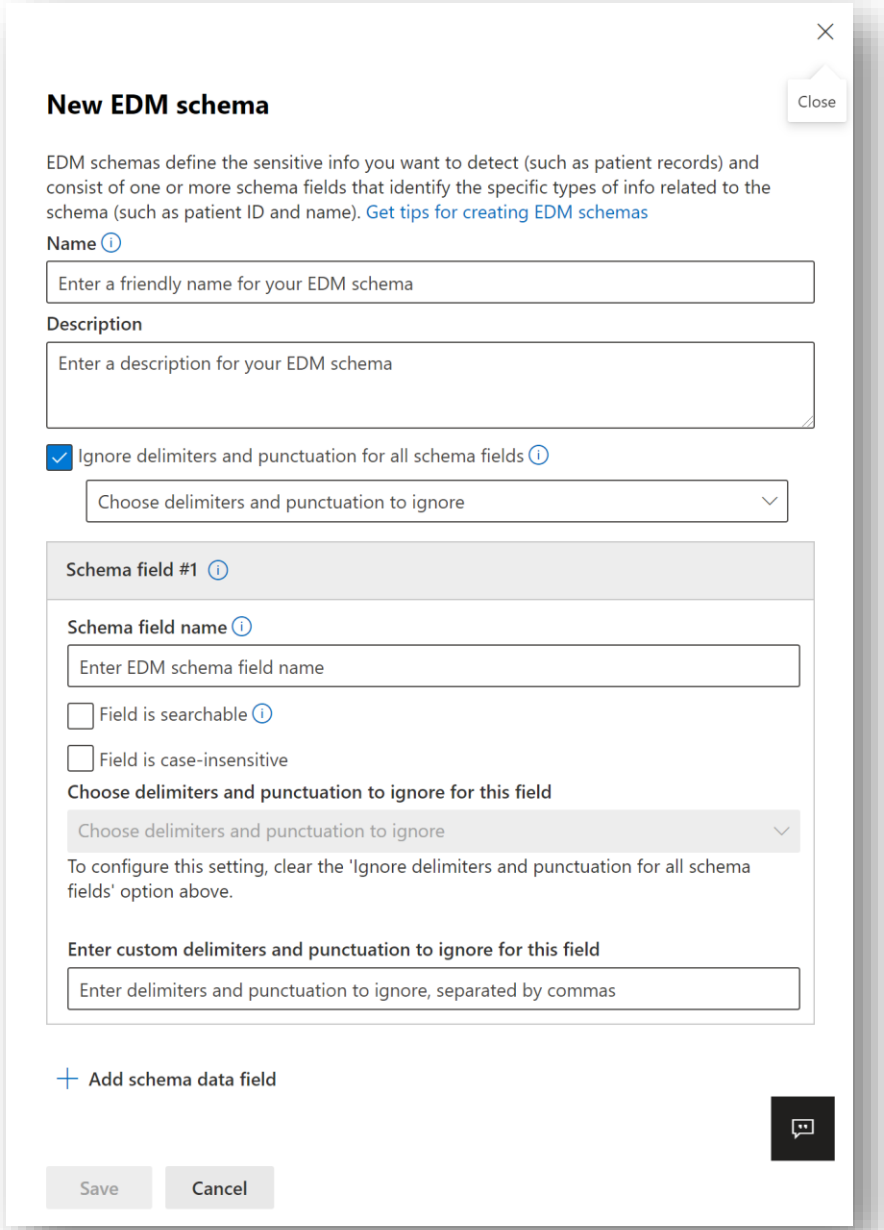
```
$edmSchemaXml=Get-Content .\edm.xml -Encoding Byte -ReadCount 0
New-DlpEdmSchema -FileData $edmSchemaXml
```

```
Get-DLPEdmSchema
```

IMPORTANT: Connect to Compliance Center PowerShell session

Define EDM Schema (the easier way)

- Open the Compliance Center
- Navigate to Data Classification
- Click on Exact Data Match
- Select EDM Schemas, create new schema
- Configure the schema
- Go drink a coffee (it can take up to one hour for the schema to become available for use)



Close

New EDM schema

EDM schemas define the sensitive info you want to detect (such as patient records) and consist of one or more schema fields that identify the specific types of info related to the schema (such as patient ID and name). Get tips for creating EDM schemas

Name

Enter a friendly name for your EDM schema

Description

Enter a description for your EDM schema

[✓] Ignore delimiters and punctuation for all schema fields

Choose delimiters and punctuation to ignore

Schema field #1

Schema field name

Enter EDM schema field name

[ ] Field is searchable

[ ] Field is case-insensitive

Choose delimiters and punctuation to ignore for this field

Choose delimiters and punctuation to ignore

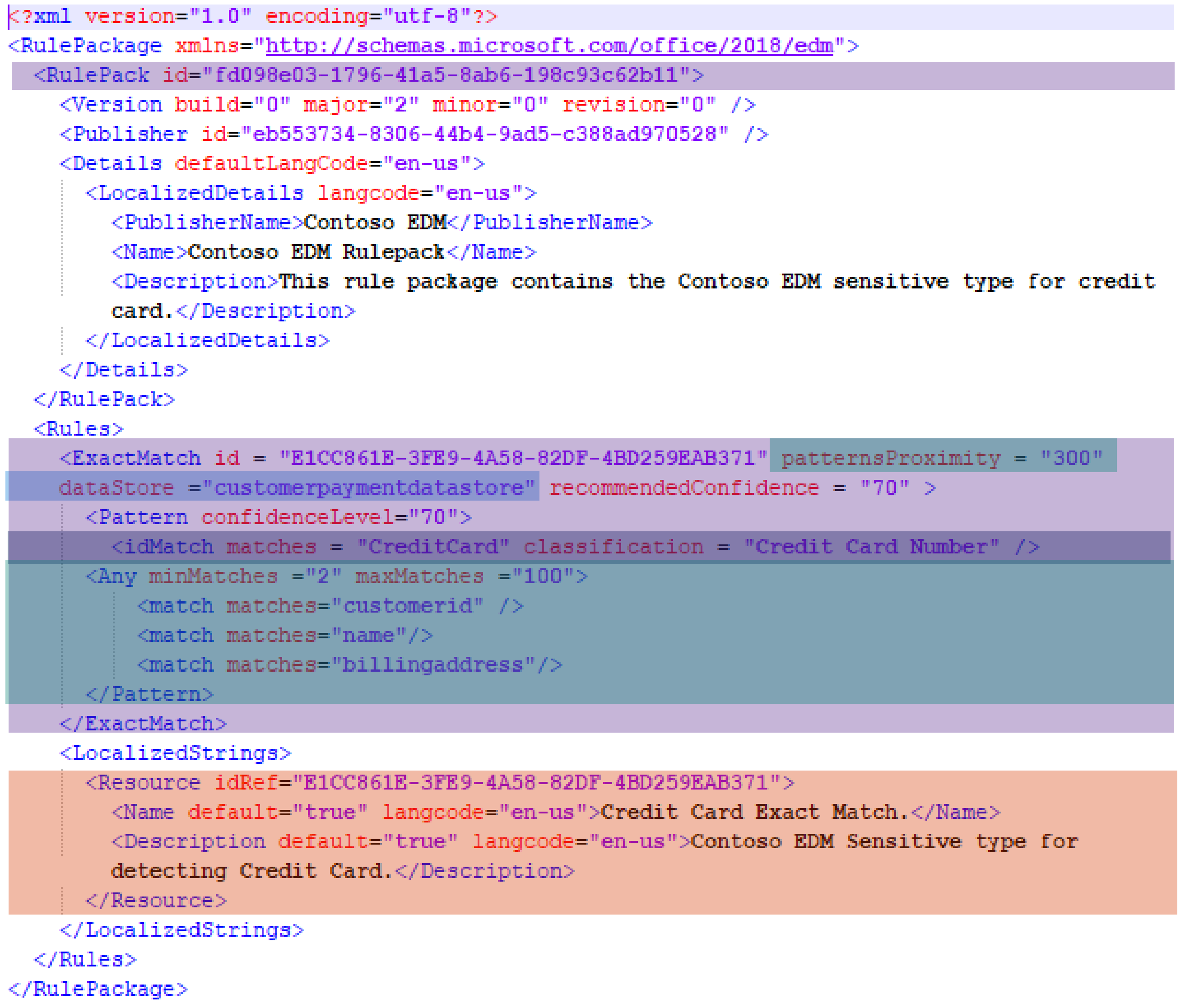
To configure this setting, clear the 'Ignore delimiters and punctuation for all schema fields' option above.

Enter custom delimiters and punctuation to ignore for this field

Enter delimiters and punctuation to ignore, separated by commas

Define EDM Sensitive Information type (the hard way)

- RulePack id & ExactMatch id
- DataStore
- idMatch
  - Matches
  - Classification
- Match
- Resources
  - idRef
  - Name & Description



```xml
<?xml version="1.0" encoding="utf-8"?>
<RulePackage xmlns="http://schemas.microsoft.com/office/2018/edm">
  <RulePack id="fd098e03-1796-41a5-8ab6-198c93c62b11">
    <Version build="0" major="2" minor="0" revision="0" />
    <Publisher id="eb553734-8306-44b4-9ad5-c388ad970528" />
    <Details defaultLangCode="en-us">
      <LocalizedDetails langcode="en-us">
        <PublisherName>Contoso EDM</PublisherName>
        <Name>Contoso EDM Rulepack</Name>
        <Description>This rule package contains the Contoso EDM sensitive type for credit
        card.</Description>
      </LocalizedDetails>
    </Details>
  </RulePack>
  <Rules>
    <ExactMatch id = "E1CC861E-3FE9-4A58-82DF-4BD259EAB371" patternsProximity = "300"
    dataStore ="customerpaymentdatastore" recommendedConfidence = "70" >
      <Pattern confidenceLevel="70">
        <idMatch matches = "CreditCard" classification = "Credit Card Number" />
      <Any minMatches ="2" maxMatches ="100">
          <match matches="customerid" />
          <match matches="name"/>
          <match matches="billingaddress"/>
      </Pattern>
    </ExactMatch>
    <LocalizedStrings>
      <Resource idRef="E1CC861E-3FE9-4A58-82DF-4BD259EAB371">
        <Name default="true" langcode="en-us">Credit Card Exact Match.</Name>
        <Description default="true" langcode="en-us">Contoso EDM Sensitive type for
        detecting Credit Card.</Description>
      </Resource>
    </LocalizedStrings>
  </Rules>
</RulePackage>
```

Sample xml for EDM Sensitive type

Upload EDM Rule Pack XML

```
$rulepack=Get-Content .\rulepack.xml -Encoding Byte -ReadCount 0
New-DlpSensitiveInformationTypeRulePackage -FileData $rulepack
```

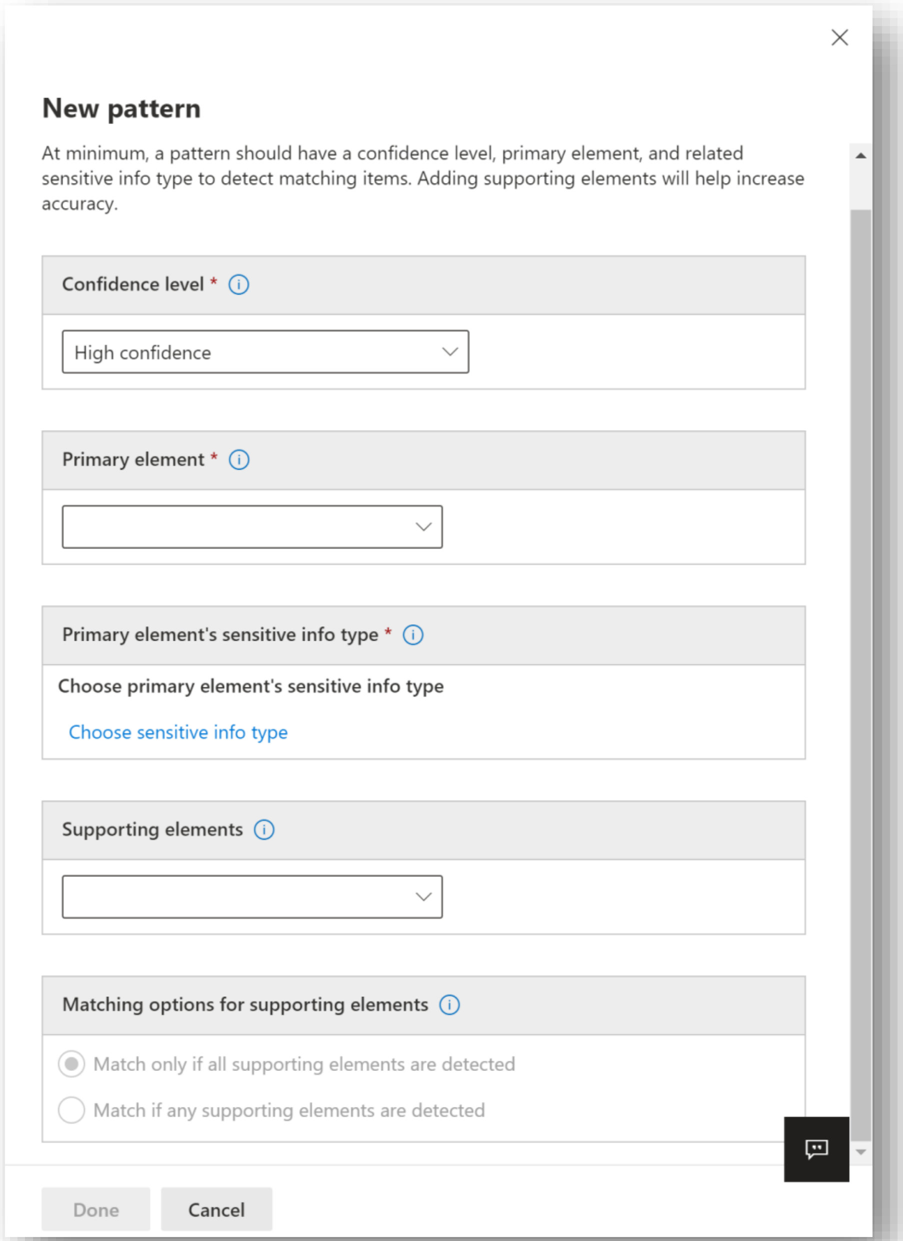
- Detail instructions on uploaded the rule pack can be found here:
  https://docs.microsoft.com/en-us/office365/securitycompliance/create-a-custom-sensitive-information-type-in-scc-powershell

- Use the following cmdlets to validate:
  Get-DLPSensitiveInformationTypeRulePackage
  Get-DLPSensitiveInformationType

IMPORTANT: Connect PS to Security & Compliance Center

Define an EDM SIT (the slightly easier way)

- In the Exact Data Match section of the Compliance Center, select EDM Sensitive Types
- Create a new EDM type
- Select your schema
- Create one or more patterns
    - Select primary element (column)
    - Select a SIT that describes it
    - Select columns to use as secondary element
    - Define matching rules (e.g. one of n, all, etc.)



New pattern

At minimum, a pattern should have a confidence level, primary element, and related sensitive info type to detect matching items. Adding supporting elements will help increase accuracy.

Confidence level *

High confidence

Primary element *

Primary element's sensitive info type *

Choose primary element's sensitive info type

Choose sensitive info type

Supporting elements

Matching options for supporting elements

[✓ Match only if all supporting elements are detected]

Match if any supporting elements are detected

Done Cancel

Hash and upload your sensitive data

EDM Upload Agent Purpose:

To one-way hash the data to have a file to upload

To Upload file with hashes to the service – where it is stored and ready for lookups

Uploading the file also upload the (automatically generated or manually entered) salt used for hashing

Set up the security group and user account

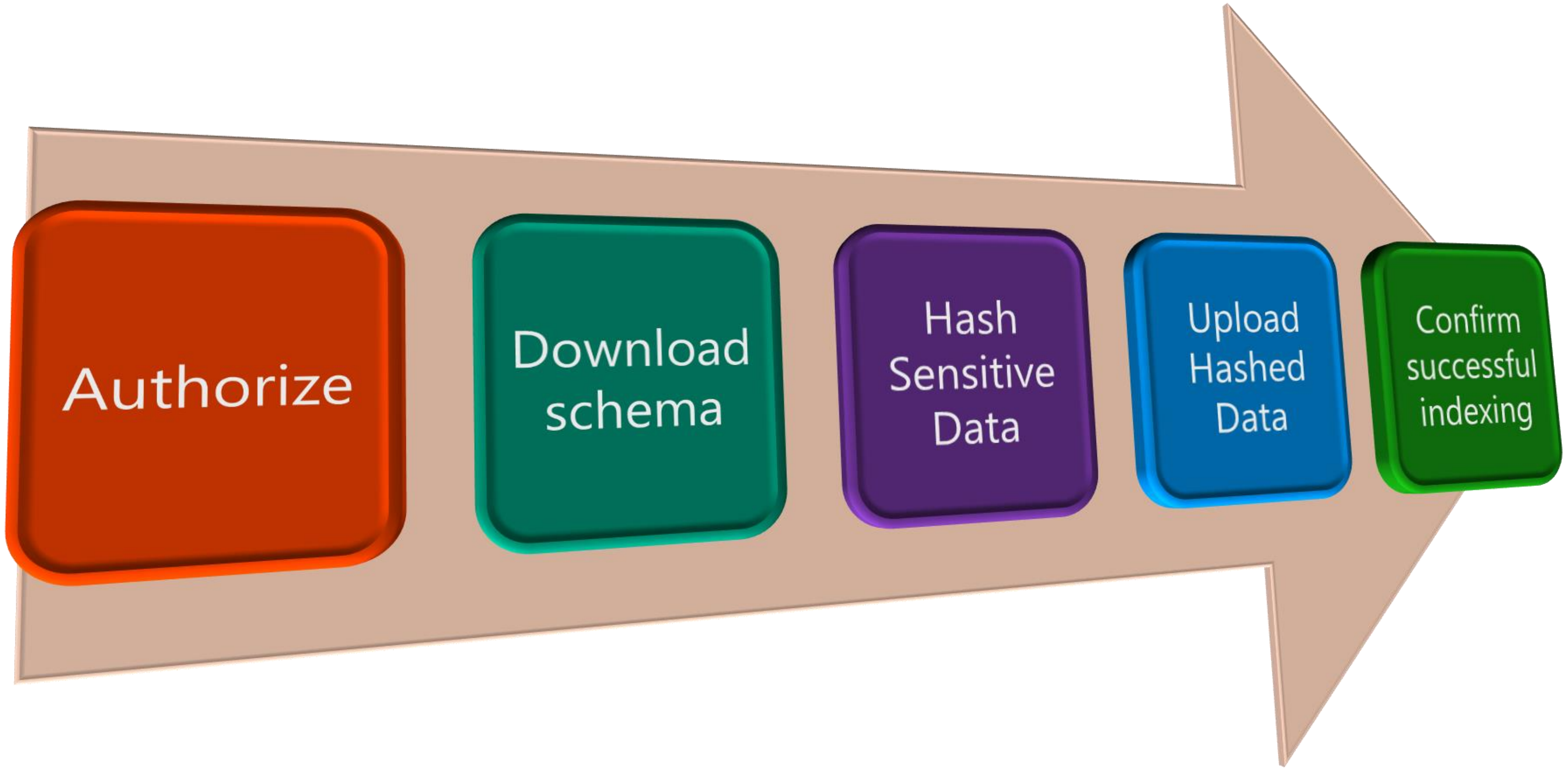
- As a global administrator, go to the admin center create a security group: EDM_DataUploaders.
- Add one or more users to the EDM_DataUploaders security group.

- ## Set up the EDM Upload Agent
  - Download EDM upload agent
  - https://go.microsoft.com/fwlink/?linkid=2088639
  - Agent trace logs located:

  C:\Program Files\Microsoft\EdmUploadAgent\TraceLogs

  Tip: DO NOT install in default folder (program files), use a custom folder, so you do not need admin privileges in the machine to use it.

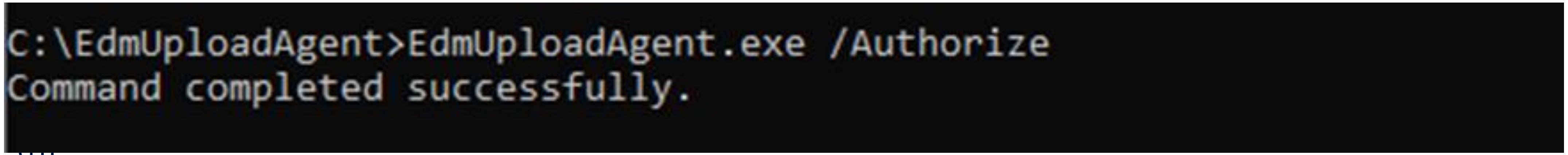
Agent Workflow



Authorize Agent

`EdmUploadAgent.exe /Authorize`

Example:



```
C:\EdmUploadAgent>EdmUploadAgent.exe /Authorize
Command completed successfully.
```

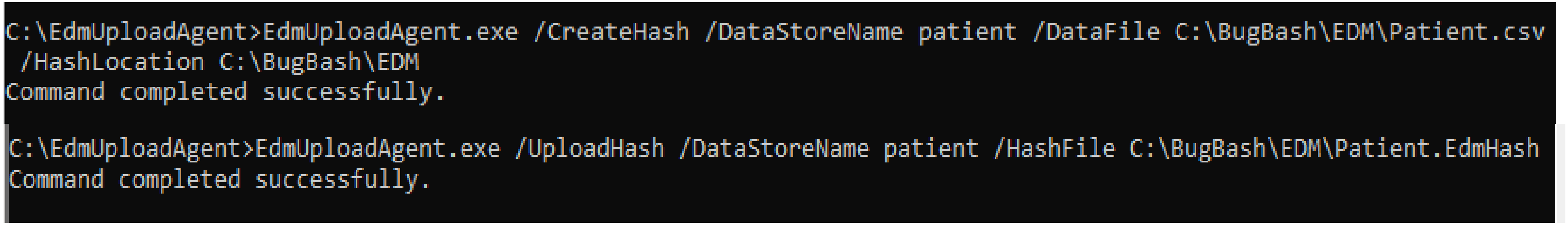
Details:

- This will prompt for user credentials to authorize the EDM upload agent to act on behalf of the user.
- It is recommended to create a separate dedicated user with minimal privileges which can be used for EDM Upload agent.*
- Authorization must be done every 30 days (depending on your tenant's AAD auth token configuration)
- Re-run the Authorize command, if any other command fails with authorization errors.
- Please note: there's an Authorize.ps1 script you can use to pass credentials interactively or script so you can pass them as a SecureString

Hash and upload Sensitive Data

```
EdmUploadAgent.exe /CreateHash /DataStoreName <DataStoreName> /DataFile <DataFilePath> /HashLocation <HashedFileLocation>
EdmUploadAgent.exe /UploadHash /DataStoreName <DataStoreName> /HashFile <HashedSourceFilePath>
```

Example:



```
C:\EdmUploadAgent>EdmUploadAgent.exe /CreateHash /DataStoreName patient /DataFile C:\BugBash\EDM\Patient.csv
 /HashLocation C:\BugBash\EDM
Command completed successfully.

C:\EdmUploadAgent>EdmUploadAgent.exe /UploadHash /DataStoreName patient /HashFile C:\BugBash\EDM\Patient.EdmHash
Command completed successfully.
```

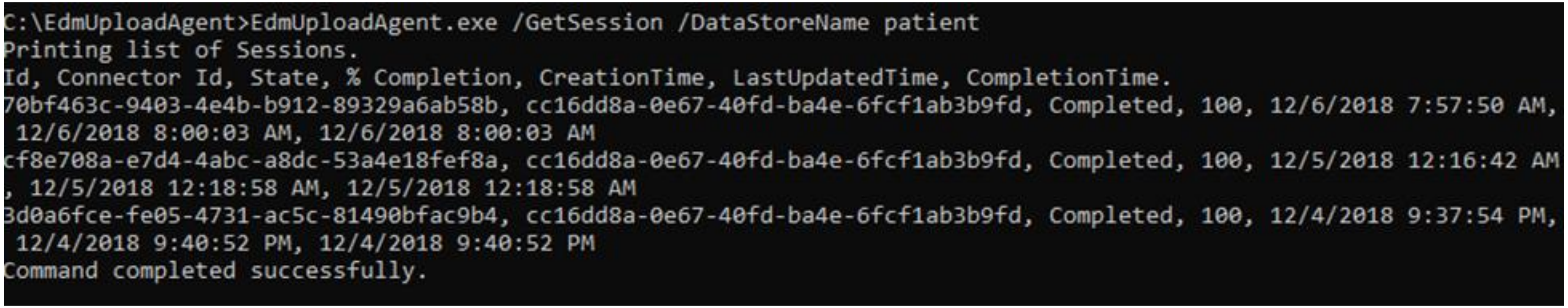
Details:

- DataStoreName: The name of the data store whose schema has already been defined. Hint: same name as the schema.
- DataFile: Provide the full path to the data file.
- HashLocation: Provide the path to the folder where the hash file should be created.
- The naming format of the hash file created is “datafilename.EdmHash”.

List Upload Sessions and check status

```
EdmUploadAgent.exe /GetSession /DataStoreName <DataStoreName>
```

Example:



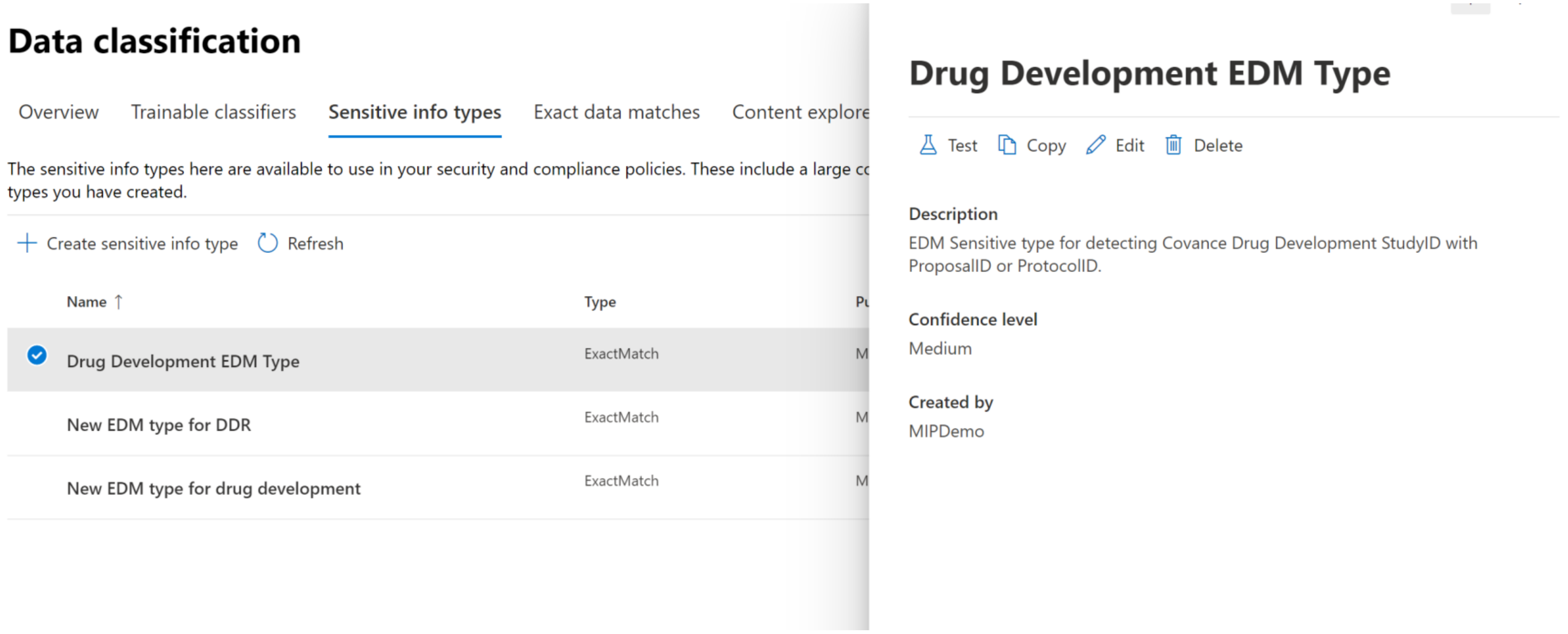
```
C:\EdmUploadAgent>EdmUploadAgent.exe /GetSession /DataStoreName patient
Printing list of Sessions.
Id, Connector Id, State, % Completion, CreationTime, LastUpdatedTime, CompletionTime.
70bf463c-9403-4e4b-b912-89329a6ab58b, cc16dd8a-0e67-40fd-ba4e-6fcf1ab3b9fd, Completed, 100, 12/6/2018 7:57:50 AM,
 12/6/2018 8:00:03 AM, 12/6/2018 8:00:03 AM
cf8e708a-e7d4-4abc-a8dc-53a4e18fef8a, cc16dd8a-0e67-40fd-ba4e-6fcf1ab3b9fd, Completed, 100, 12/5/2018 12:16:42 AM
, 12/5/2018 12:18:58 AM, 12/5/2018 12:18:58 AM
3d0a6fce-fe05-4731-ac5c-81490bfac9b4, cc16dd8a-0e67-40fd-ba4e-6fcf1ab3b9fd, Completed, 100, 12/4/2018 9:37:54 PM,
 12/4/2018 9:40:52 PM, 12/4/2018 9:40:52 PM
Command completed successfully.
```

Details:

- Lists all the data uploads done for a specific data store.
- Note: until the third column says "Completed" the table is not ready for use. Can take *hours* to complete indexing and replication for large tables.

Testing EDM Sensitive Info Type

- Classifications, Sensitive info type
- Select EDM type
- Upload file to test



Data classification

Overview | Trainable classifiers | **Sensitive info types** | Exact data matches | Content explore

The sensitive info types here are available to use in your security and compliance policies. These include a large co
types you have created.

+ Create sensitive info type   Refresh

Name ↑	Type	Pu
✓ Drug Development EDM Type	ExactMatch	M
New EDM type for DDR	ExactMatch	M
New EDM type for drug development	ExactMatch	M

Drug Development EDM Type

Test Copy Edit Delete

Description
EDM Sensitive type for detecting Covance Drug Development StudyID with ProposalID or ProtocolID.

Confidence level
Medium

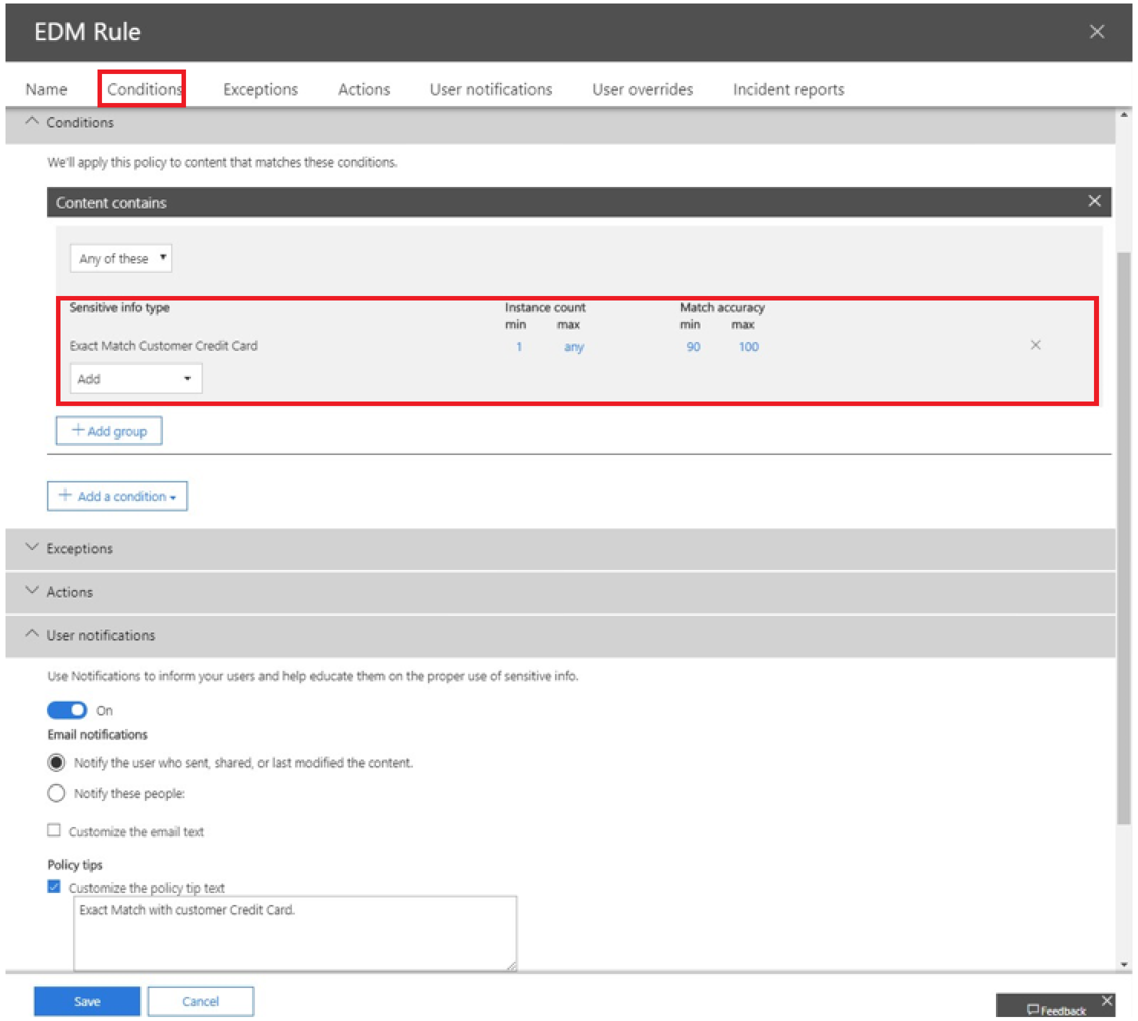
Created by
MIPDemo

Notes:

- This is in the regular SIT UI, not in the EDM UI!!!
- EDM sensitive info type changes take up to ***one hour*** to be propagated. You might be testing the old version!!!

Create a DLP Policy that uses EDM

- Use DLP Policy wizard to create policy and corresponding rule
  - Conditions:
    - Select the custom EDM sensitive type you just created
  - Actions:
    - Only user notification (policy tip) in OWA is supported.
    - DLP block actions for Exchange, SharePoint and OneDrive planned.
  - User Notification:
    - Turn on policy tip



EDM Rule

Name | Conditions | Exceptions | Actions | User notifications | User overrides | Incident reports

Conditions

We'll apply this policy to content that matches these conditions.

Content contains

Any of these

Sensitive info type	Instance count min	Instance count max	Match accuracy min	Match accuracy max
Exact Match Customer Credit Card	1	any	90	100

Add

+ Add group

+ Add a condition

Exceptions

Actions

User notifications

Use Notifications to inform your users and help educate them on the proper use of sensitive info.

On

Email notifications

(●) Notify the user who sent, shared, or last modified the content.

( ) Notify these people:

☐ Customize the email text

Policy tips

☑ Customize the policy tip text

Exact Match with customer Credit Card.

Save | Cancel | Feedback

Define EDM Schema (the hard way)

```xml
<?xml version="1.0" encoding="utf-8"?>
<EdmSchema xmlns="http://schemas.microsoft.com/office/2018/edm">
  <DataStore name="SampleDataStore" description="Sample Datastore" version="1">
    <Field name="id" unique="true" searchable="true" />
    <Field name="firstname" unique="false" searchable="true" />
    <Field name="lastname" unique="false" searchable="false" />
    <Field name="title" unique="false" searchable="false" />
    <Field name="dob" unique="false" searchable="false" />
    <Field name="creditcard" unique="false" searchable="true" />
    <Field name="ssn" unique="false" searchable="true" />
  </DataStore>
</EdmSchema>
```

Sample EDM Schema xml

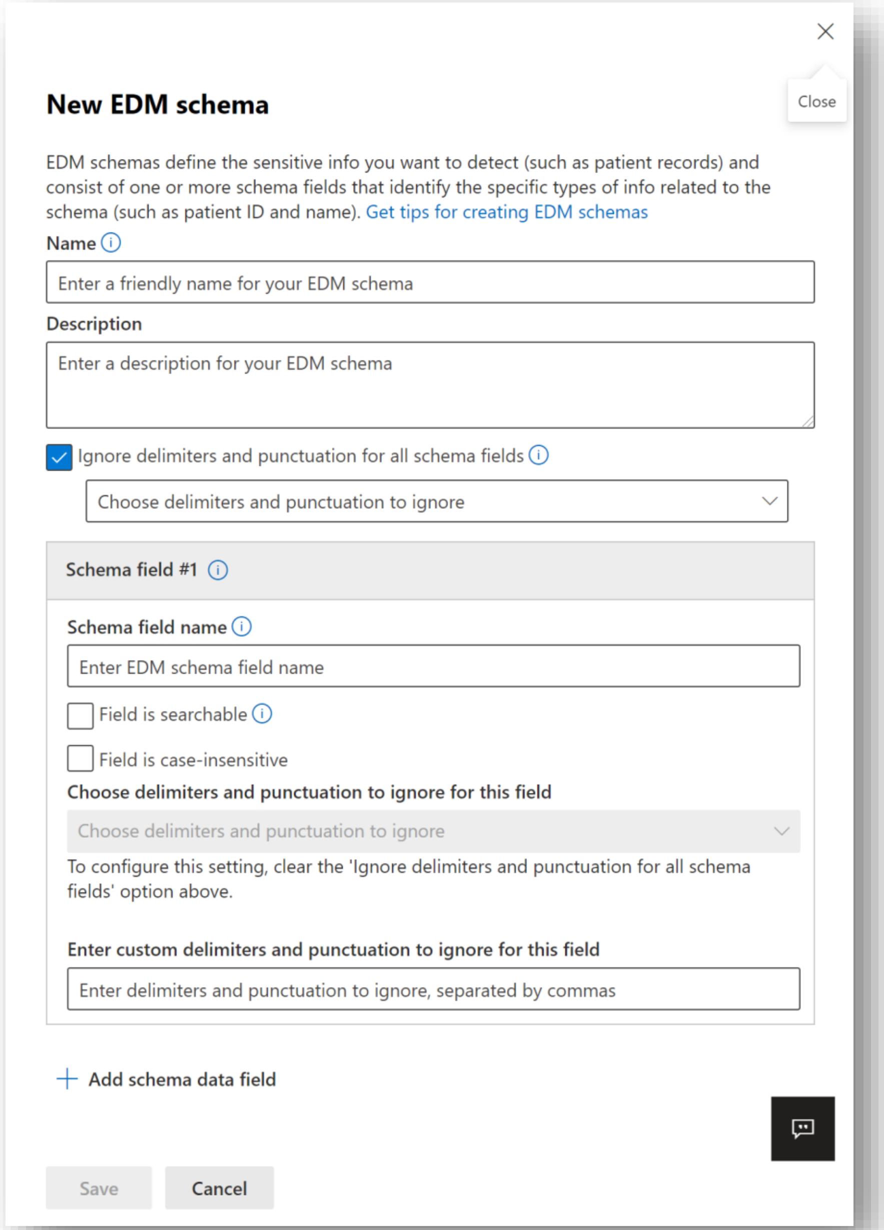
```
$edmSchemaXml=Get-Content .\edm.xml -Encoding Byte -ReadCount 0
New-DlpEdmSchema -FileData $edmSchemaXml

Get-DLPEdmSchema
```

IMPORTANT: Connect to Compliance Center PowerShell session

Define EDM Schema (the easy way)

- Open the Compliance Center
- Navigate to Data Classification
- Click on Exact Data Match
- Select EDM Schemas, create new schema
- Configure the schema
- Go drink a coffee (it can take up to one hour for the schema to become available for use)



New EDM schema

EDM schemas define the sensitive info you want to detect (such as patient records) and consist of one or more schema fields that identify the specific types of info related to the schema (such as patient ID and name). Get tips for creating EDM schemas

Name

Enter a friendly name for your EDM schema

Description

Enter a description for your EDM schema

[✓] Ignore delimiters and punctuation for all schema fields

Choose delimiters and punctuation to ignore

Schema field #1

Schema field name

Enter EDM schema field name

[ ] Field is searchable

[ ] Field is case-insensitive

Choose delimiters and punctuation to ignore for this field

Choose delimiters and punctuation to ignore

To configure this setting, clear the 'Ignore delimiters and punctuation for all schema fields' option above.

Enter custom delimiters and punctuation to ignore for this field

Enter delimiters and punctuation to ignore, separated by commas

Define EDM Sensitive Information type (the hard way)

- RulePack id & ExactMatch id
- DataStore
- idMatch
  - Matches
  - Classification
- Match
- Resources
  - idRef
  - Name & Description

```xml
<?xml version="1.0" encoding="utf-8"?>
<RulePackage xmlns="http://schemas.microsoft.com/office/2018/edm">
  <RulePack id="fd098e03-1796-41a5-8ab6-198c93c62b11">
    <Version build="0" major="2" minor="0" revision="0" />
    <Publisher id="eb553734-8306-44b4-9ad5-c388ad970528" />
    <Details defaultLangCode="en-us">
      <LocalizedDetails langcode="en-us">
        <PublisherName>Contoso EDM</PublisherName>
        <Name>Contoso EDM Rulepack</Name>
        <Description>This rule package contains the Contoso EDM sensitive type for credit
        card.</Description>
      </LocalizedDetails>
    </Details>
  </RulePack>
  <Rules>
    <ExactMatch id = "E1CC861E-3FE9-4A58-82DF-4BD259EAB371" patternsProximity = "300"
    dataStore ="customerpaymentdatastore" recommendedConfidence = "70" >
      <Pattern confidenceLevel="70">
        <idMatch matches = "CreditCard" classification = "Credit Card Number" />
      <Any minMatches ="2" maxMatches ="100">
          <match matches="customerid" />
          <match matches="name"/>
          <match matches="billingaddress"/>
      </Pattern>
    </ExactMatch>
    <LocalizedStrings>
      <Resource idRef="E1CC861E-3FE9-4A58-82DF-4BD259EAB371">
        <Name default="true" langcode="en-us">Credit Card Exact Match.</Name>
        <Description default="true" langcode="en-us">Contoso EDM Sensitive type for
        detecting Credit Card.</Description>
      </Resource>
    </LocalizedStrings>
  </Rules>
</RulePackage>
```

Sample xml for EDM Sensitive type

Upload EDM Rule Pack XML

```
$rulepack=Get-Content .\rulepack.xml -Encoding Byte -ReadCount 0
New-DlpSensitiveInformationTypeRulePackage -FileData $rulepack
```

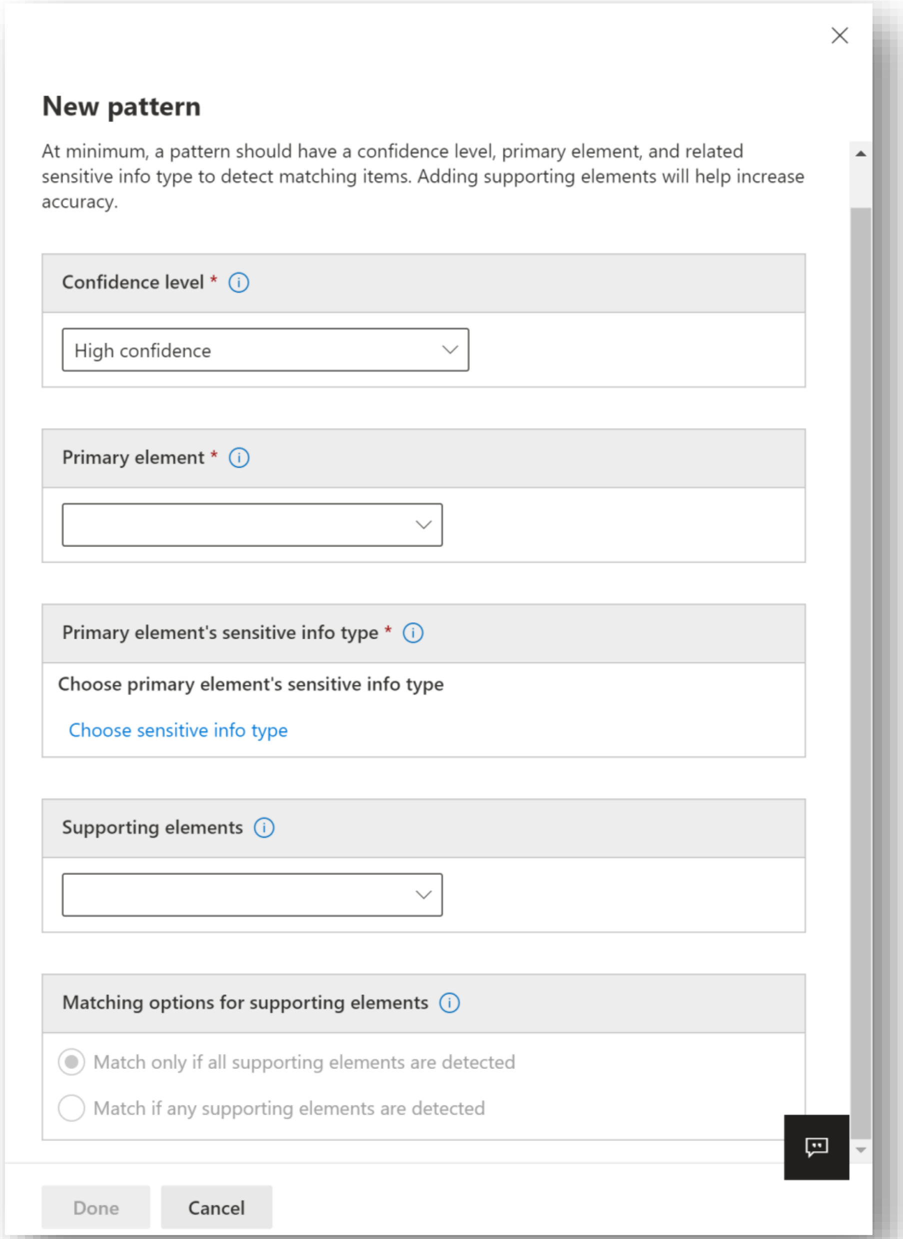
- Detail instructions on uploaded the rule pack can be found here:
  https://docs.microsoft.com/en-us/office365/securitycompliance/create-a-custom-sensitive-information-type-in-scc-powershell

- Use the following cmdlets to validate:
  Get-DLPSensitiveInformationTypeRulePackage
  Get-DLPSensitiveInformationType

IMPORTANT: Connect PS to Security & Compliance Center

Define EDM Sensitive Information type (the easy way)

- In the Exact Data Match section of the Compliance Center, select EDM Sensitive Types
- Create a new EDM type
- Select your schema
- Create one or more patterns



New pattern

At minimum, a pattern should have a confidence level, primary element, and related sensitive info type to detect matching items. Adding supporting elements will help increase accuracy.

Confidence level *

High confidence

Primary element *

Primary element's sensitive info type *

Choose primary element's sensitive info type

Choose sensitive info type

Supporting elements

Matching options for supporting elements

[✓ Match only if all supporting elements are detected]

Match if any supporting elements are detected

Hash and upload your sensitive data

EDM Upload Agent Purpose:

To one-way hash the data for upload

Upload Hashed file to the service – where it is stored and ready for lookups

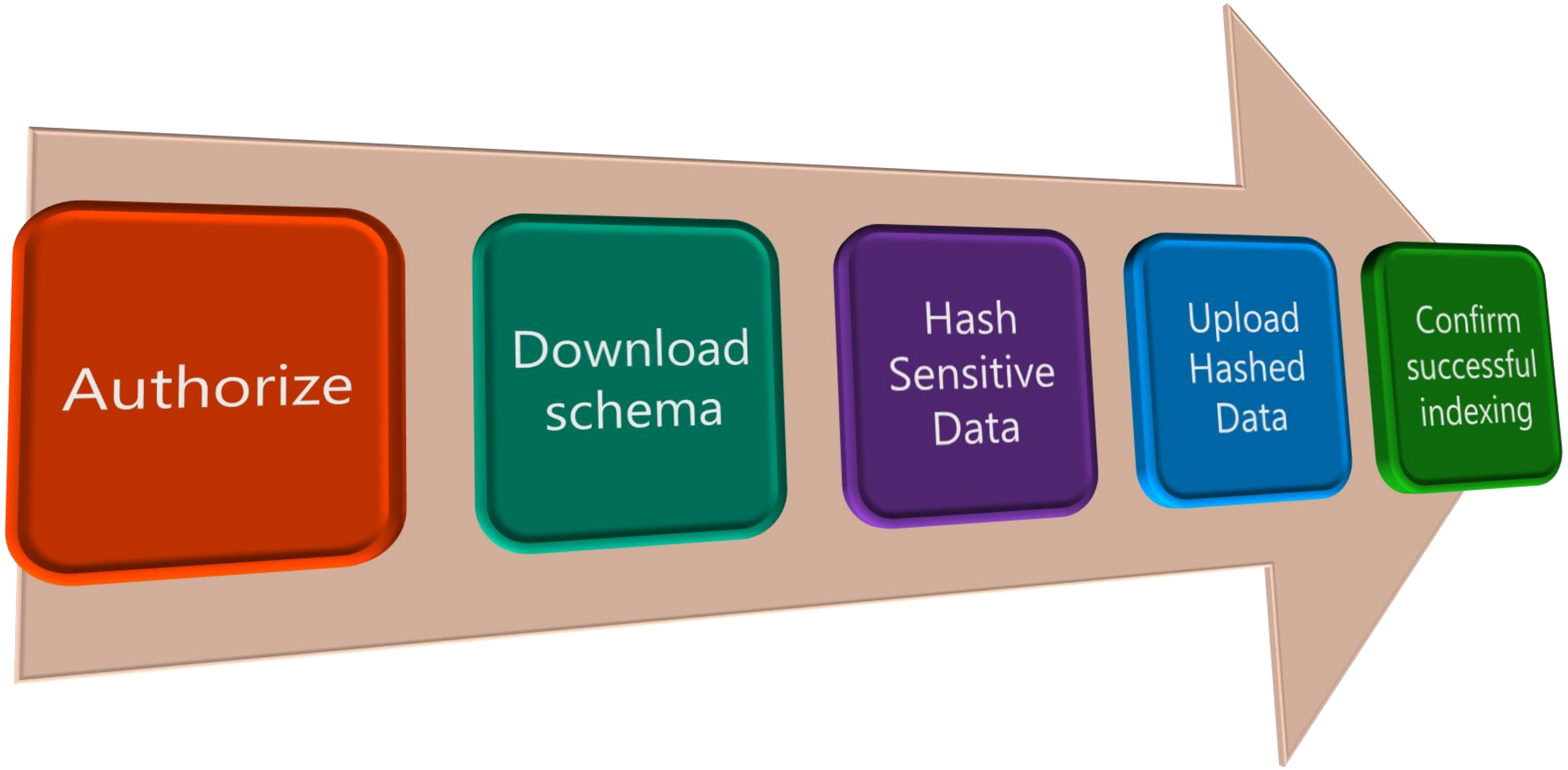
Set up the security group and user account

- As a global administrator, go to the admin center create a security group: EDM_DataUploaders.
- Add one or more users to the EDM_DataUploaders security group.
- Make sure user is a local admin on the machine where EDM Upload Agent is installed.

- ### Set up the EDM Upload Agent
  - Download EDM upload agent
  - https://go.microsoft.com/fwlink/?linkid=2088639
  - Agent trace logs located:

  C:\Program Files\Microsoft\EdmUploadAgent\TraceLogs

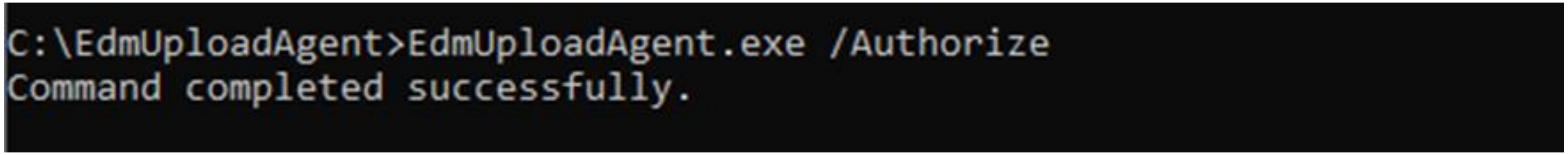
Agent Workflow



Authorize Agent

`EdmUploadAgent.exe /Authorize`

Example:



```
C:\EdmUploadAgent>EdmUploadAgent.exe /Authorize
Command completed successfully.
```

Details:

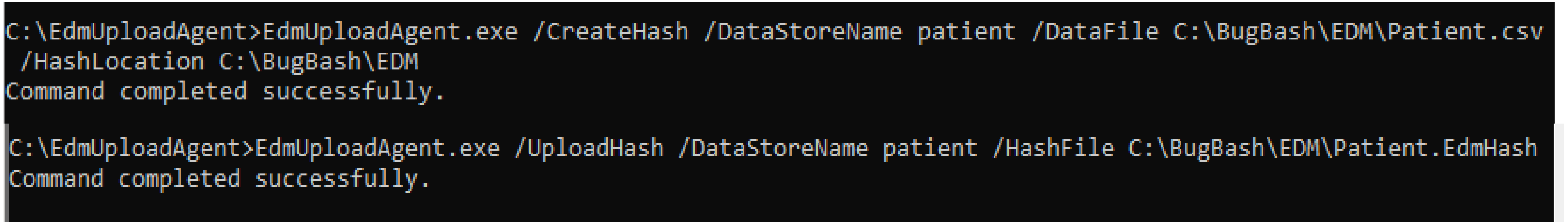
- This will prompt for user credentials to authorize the EDM upload agent to act on behalf of the user.
- It is recommended to create a separate dedicated tenant user with minimal privileges which can be used for EDM Upload agent.*
- Re-run the Authorize command, if any other command fails with authorization errors.

IMPORTANT: Launch Command Prompt as Administrator

Hash and upload Sensitive Data

```
EdmUploadAgent.exe /CreateHash /DataStoreName <DataStoreName>
/DataFile <DataFilePath> /HashLocation <HashedFileLocation>
EdmUploadAgent.exe /UploadHash /DataStoreName <DataStoreName>
/HashFile <HashedSourceFilePath>
```

Example:



```
C:\EdmUploadAgent>EdmUploadAgent.exe /CreateHash /DataStoreName patient /DataFile C:\BugBash\EDM\Patient.csv
 /HashLocation C:\BugBash\EDM
Command completed successfully.

C:\EdmUploadAgent>EdmUploadAgent.exe /UploadHash /DataStoreName patient /HashFile C:\BugBash\EDM\Patient.EdmHash
Command completed successfully.
```

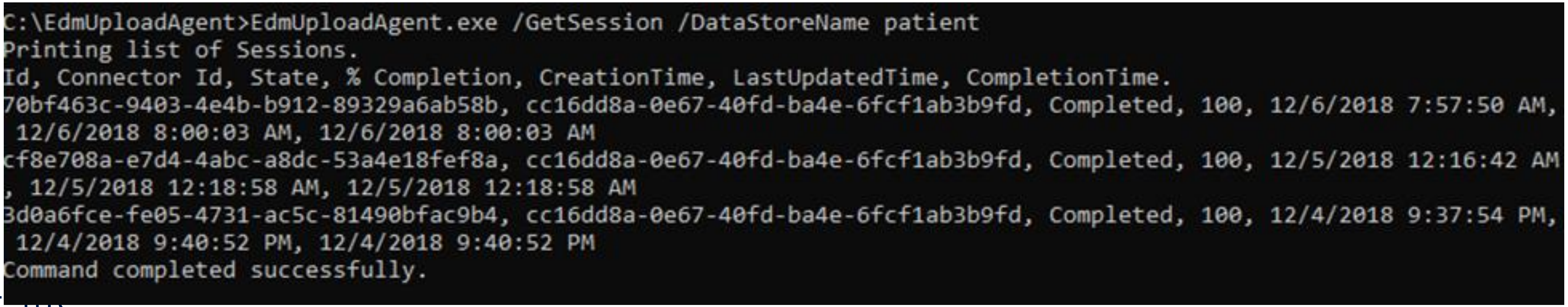
Details:

- DataStoreName: The name of the data store whose schema has already been defined.
- DataFile: Provide the full path to the data file.
- HashLocation: Provide the path to the folder where the hash file should be created.
- The naming format of the hash file created is “datafilename.EdmHash”.

List Upload Sessions and check status

```
EdmUploadAgent.exe /GetSession /DataStoreName <DataStoreName>
```

Example:



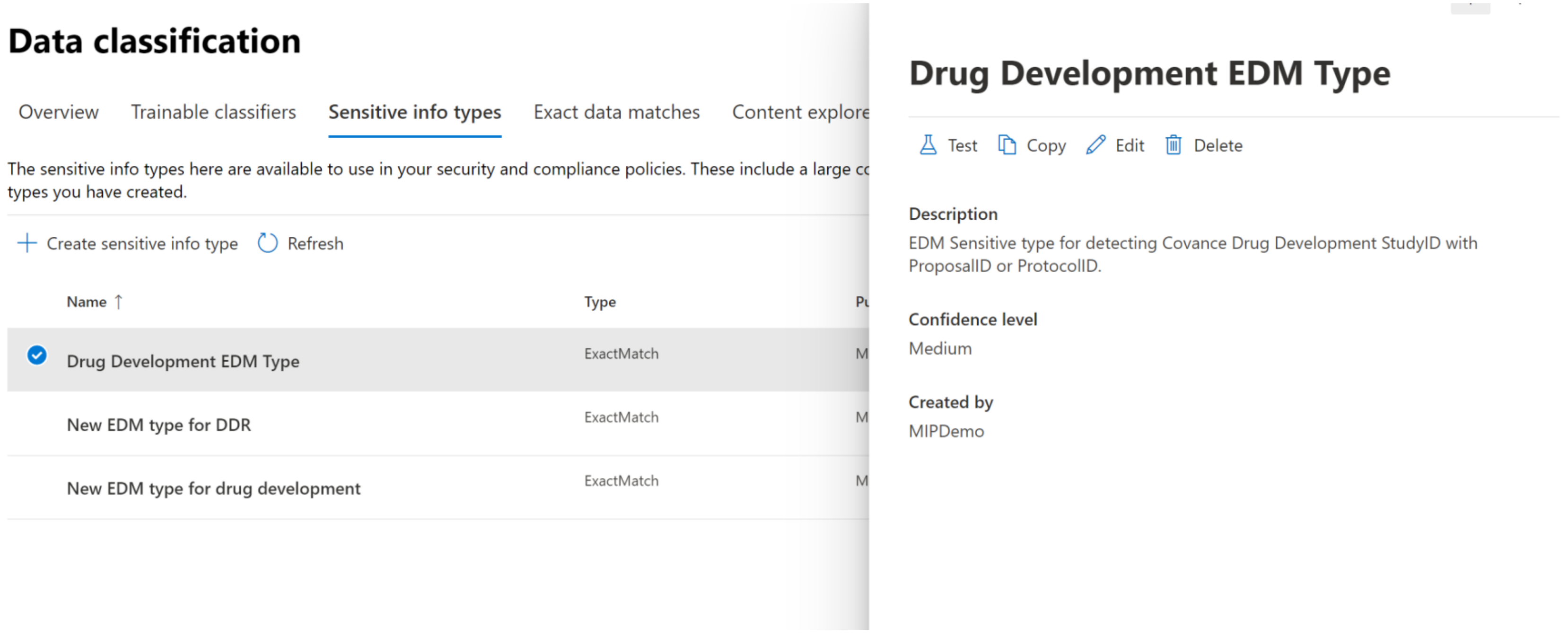
```
C:\EdmUploadAgent>EdmUploadAgent.exe /GetSession /DataStoreName patient
Printing list of Sessions.
Id, Connector Id, State, % Completion, CreationTime, LastUpdatedTime, CompletionTime.
70bf463c-9403-4e4b-b912-89329a6ab58b, cc16dd8a-0e67-40fd-ba4e-6fcf1ab3b9fd, Completed, 100, 12/6/2018 7:57:50 AM,
 12/6/2018 8:00:03 AM, 12/6/2018 8:00:03 AM
cf8e708a-e7d4-4abc-a8dc-53a4e18fef8a, cc16dd8a-0e67-40fd-ba4e-6fcf1ab3b9fd, Completed, 100, 12/5/2018 12:16:42 AM
, 12/5/2018 12:18:58 AM, 12/5/2018 12:18:58 AM
3d0a6fce-fe05-4731-ac5c-81490bfac9b4, cc16dd8a-0e67-40fd-ba4e-6fcf1ab3b9fd, Completed, 100, 12/4/2018 9:37:54 PM,
 12/4/2018 9:40:52 PM, 12/4/2018 9:40:52 PM
Command completed successfully.
```

Details:

- Lists all the data uploads done for a specific data store.

Testing EDM Sensitive Info Type

- Classifications, Sensitive info type
- Select EDM type
- Upload file to test



Data classification

Overview | Trainable classifiers | **Sensitive info types** | Exact data matches | Content explore

The sensitive info types here are available to use in your security and compliance policies. These include a large co
types you have created.

+ Create sensitive info type   Refresh

Name ↑	Type	Pu
✓ Drug Development EDM Type	ExactMatch	M
New EDM type for DDR	ExactMatch	M
New EDM type for drug development	ExactMatch	M

Drug Development EDM Type

Test   Copy   Edit   Delete

Description
EDM Sensitive type for detecting Covance Drug Development StudyID with ProposalID or ProtocolID.

Confidence level
Medium

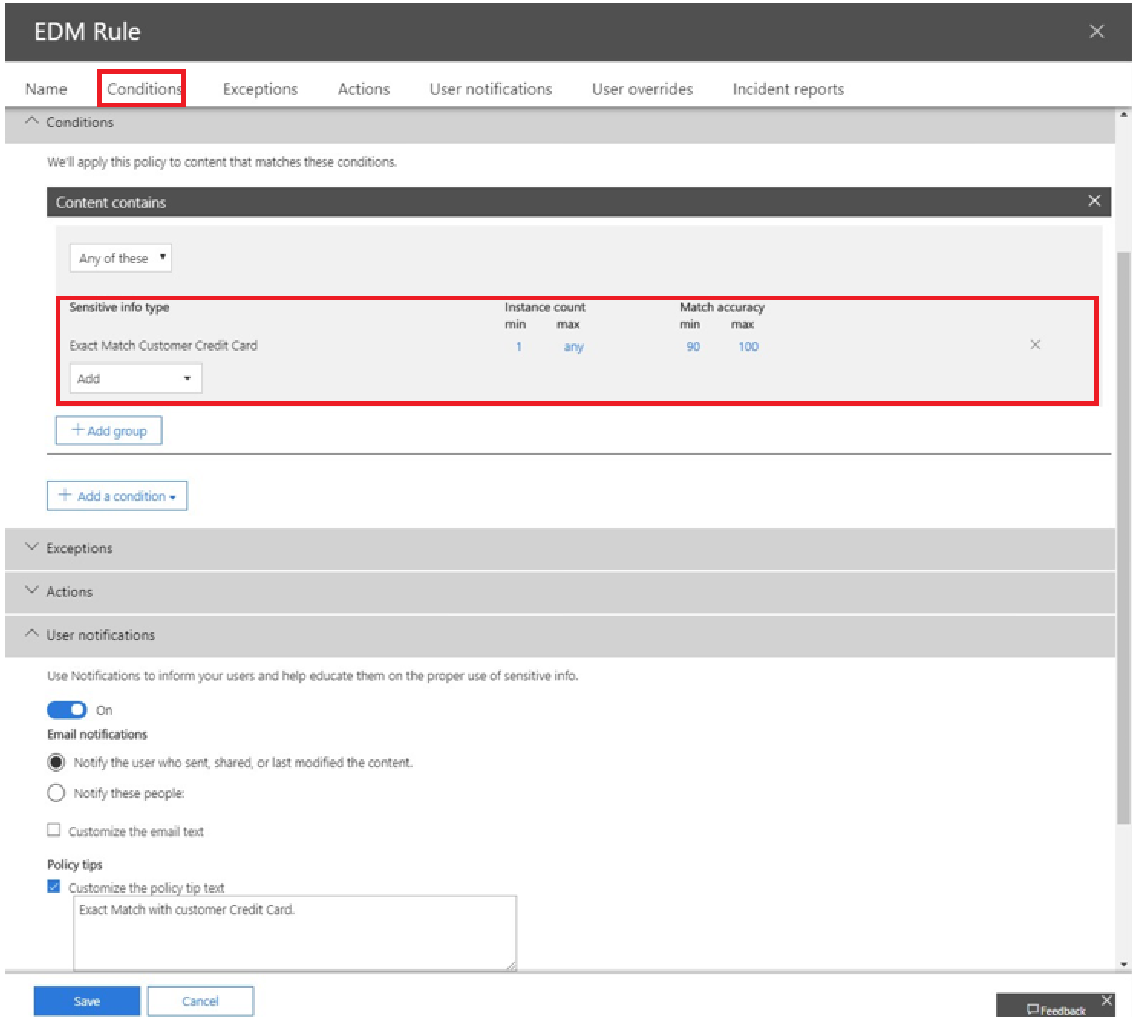
Created by
MIPDemo

Notes:

- This is in the regular SIT UI, not in the EDM UI!!!
- EDM sensitive info type changes take up to one hour to be propagated. You might be testing the old version!!!

Create a DLP Policy that uses EDM

- Use DLP Policy wizard to create policy and corresponding rule
  - Conditions:
    - Select the custom EDM sensitive type you just created
  - Actions:
    - Only user notification (policy tip) in OWA is supported.
    - DLP block actions for Exchange, SharePoint and OneDrive planned.
  - User Notification:
    - Turn on policy tip



Conditions

We'll apply this policy to content that matches these conditions.

Content contains

Any of these

Sensitive info type	Instance count min	Instance count max	Match accuracy min	Match accuracy max
Exact Match Customer Credit Card	1	any	90	100

User notifications

Use Notifications to inform your users and help educate them on the proper use of sensitive info.

On

Email notifications

- [✓ Notify the user who sent, shared, or last modified the content.]
- Notify these people:
- ☐ Customize the email text

Policy tips

- ☑ Customize the policy tip text

Exact Match with customer Credit Card.

Sneak peek: new EDM wizard

Before and after

Before:

1. Identify the structure of your data file and manually define the schema
2. Create each EDM SIT manually
3. For each EDM SIT, define each pattern manually by selecting a primary element column and additional evidence columns
4. Select the matching SIT for each primary evidence element
   - You need to ensure they match
   - You need to ensure they aren't too vague
5. You may need to use PowerShell for some advanced configurations (more on this later)
6. Hash and upload sample or production data, wait, test, make adjustments as needed.

After:

1. Upload a table with sample (fake?) data.
2. Wizard detects structure and creates schema.
3. Wizard detects matching SITs for each column and recommends primary elements.
   - Automatically validates suitability of the matching SITs to exclude most common errors.
4. Wizard creates EDM SIT with recommended patterns using the SITs.
5. Trigger SITs are tested against the data as you go.

Demo

EDM Best practices

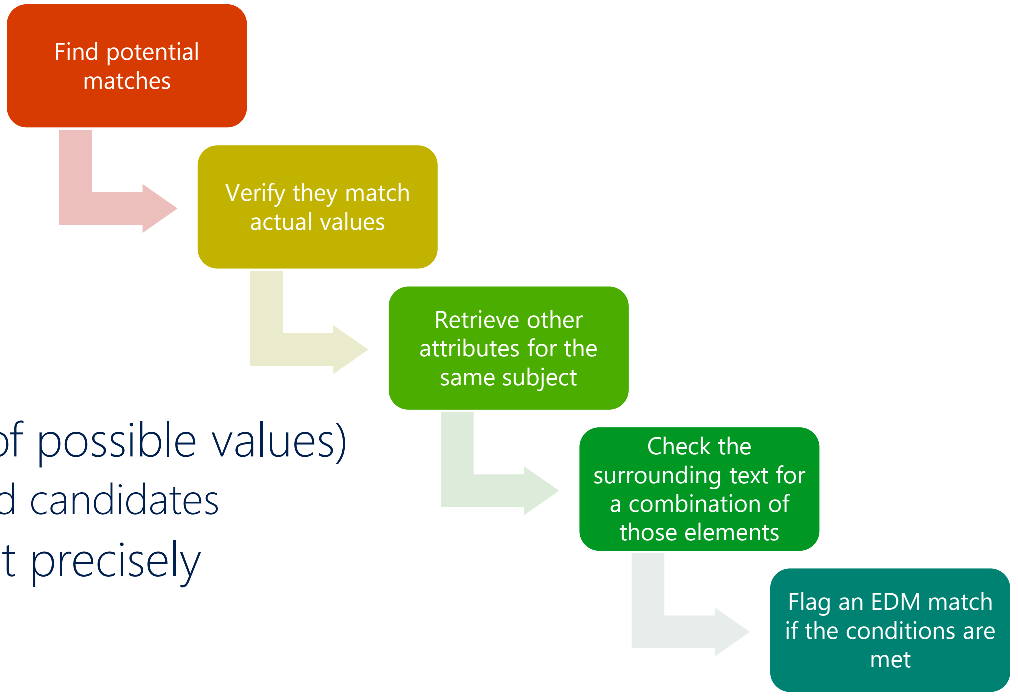
Common best practices

- 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.
- 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]`
- Install 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.
- 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!
- Look hard at your primary evidence columns: not everything that's important must be a primary element
  - E.g. a last name is important, but unlikely to be of relevance alone, so define more structured columns such as email address, account number or SSN as primary elements and put last name as a secondary element to them if possible.

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").


Find potential matches
Verify they match actual values
Retrieve other attributes for the same subject
Check the surrounding text for a combination of those elements
Flag an EDM match if the conditions are met

If you get something from this webinar, let it be NOT TO USE \b\w+\b (or even worse, \w+) as a SIT trigger!!!!

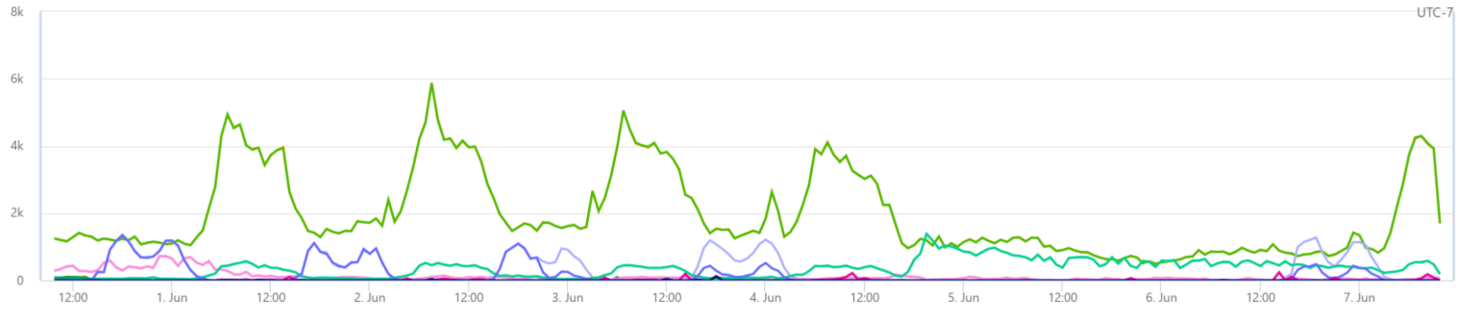
Additional considerations for the "trigger" SIT

- For each primary element you must define a SIT which will find candidates to match
- The SIT used must not be too frequently present in content
  - E.g. not in every document or email generated.
  - Keep in mind that document metadata and email headers are also scanned!
    - 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 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
- 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.

What happens if you choose the wrong SIT

- If the SIT doesn't meet the conditions before:

  1) It may cause missed detections because the system is too busy processing matches
     - If a SIT causes an average of more than ~100 matches per second in your tenant, it is likely a bad candidate to use as trigger for a primary element.
     - Missed detections are *silent*.
       You won't know what you are missing.
     - Be aware of peaks in load, plan for the worst.
     - If there's significant throttling, we may contact you.



  2) It may cause processing to timeouts if any given value matches thousands of rows in the table.
     - E.g. if you have a "birth year" column, each lookup could return hundreds of thousands of rows, each of which has to be compared to each four-digit string in each document, which could take minutes, so it will timeout and not return a match.
  3) It might not match candidates, so EDM doesn't get to try them against the table.
  4) It might match a substring or superstring of what's in the table, so the hashes won't match.
  5) It might match something that is spelled out or ordered differently from what's in the table, so the hashes won't match.

IF you get something from this webinar, let it be NOT TO USE a "wildcard" regex as a SIT trigger!!!!
Examples:

- \b\w+\b (matches every word)
- \w+ (matches every string)
- [A-Za-z0-9]+ (matches any string)
- \d* (matches any number of any length)

So I can only use things that match a strict regex?

NO!

First, you may be able to use a tricky column as “additional evidence” with another column as primary.

You can use a regular expression with additional conditions:

E.g. an email address regex that excludes those in the TO and CC lines:

```
(?m)(?<!((From|To|CC): ([a-zA-Z0-9. <>@\.\-\(\);])*))([a-zA-Z0-9.-]+\@[a-zA-Z0-9.]+\.[a-zA-Z]{2,15})\b
```

You can use a “Loose” regular expression with keywords:

E.g. \b\d{6,9}\b, but with the keywords “account number”, “account ID”, or “acct#” nearby.

You can use multiple regexes (in different patterns or as a combined regex joined by an OR condition)

E.g. you want to detect account numbers in multiple different formats: ABC12345, A12345XY, 12345678901234 , A1B2C3D123456, and more

Bas option: single regex pattern that detects all of them (and more): [A-Z\d]{8,14}

Better option: combine regexes for each format via “|”:

```
\b([A-Z]{3}\d{5})|([A-Z]\d{5}\[XY]{2}|\d{14}|([A-Z]\d){4}\d{5})\b
```

But more importantly: SIT <> REGEX, there are other options:

- Named entities
- Dictionaries
- Combined dictionaries

Using Named Entity Recognition with EDM

What is NER?

- Algorithmic classifiers that detect specific types of entities:
  - All full names
  - All physical locations
  - Physical locations in different countries
  - "All last names" and others on the roadmap
  - Etc.
- Match based on format, context and internal dictionaries.

Not often used alone. Typically used to enhance accuracy of rules that look for PII

- e.g. an SSN *and someone's name*

Can also be used with EDM as a trigger SIT for the corresponding primary element, e.g. to match a "Full Name" column.

Main limitation: names and addresses may be written in a different format than how NER captures the match. E.g. it might capture a full address including province and country, while your table might include only the street address.

Using Dictionaries

Dictionaries? Isn't EDM better?

	Dictionaries	EDM
Table size	~100K rows (1MB compressed)	100M rows
Accuracy	Single value match	Multi-column match
Case sensitivity	Case insensitive	Case sensitive or insensitive
Privacy	Need to upload the data to Microsoft	Only need to supply salted hashes of data
Max detection rate	Not limited	100s per second

EDM SIT triggered by a dictionary: the best of both worlds!

- Use dictionary to find candidates, use EDM to find the exact matches to combined columns.

But what about dictionary sizes? What if I have millions of records to protect?

- You might have 100M customers, but there aren't 100M last names.

First name	Last name	SSN
Kye	Reese	111-11-1111
John	Connor	983-87-3339
SARAH	CONNOR	292-09-2866
Peter	Silberman	283-33-2114
Ed	Traxler	091-22-9384

Deduplicate

Last name
Reese
Connor
Silberman
Traxler

Upload as dictionary

All last names
Connor
Reese
Silberman
Traxler

+ Other PII

Hash and upload

Firstname	Lastname	SSN
56f6e20696e2	6f6e2069634	0636f6d70617
4b177e0db1d	82517c9053d	9e7280c96dd
ebc4103dda7	6e206f662073	580e00857dd
56f6e20696e2	6f6e2069634	0636f6d70617

Tip: sort the dictionary before upload (makes compression more efficient)

You might also want to include additional conditions in the dictionary-based SIT

(Very) advanced scenario: Combined Dictionaries

What if you have to detect complex elements such as street addresses?

E.g. "29833 NE 29th St., Redmond, WA, 98052, USA"

Problems:

- Too many different addresses to fit in a dictionary
- Address might not be spelled out that way in a document

Solution:

- Parse the column in parts
- Define SITs (dictionary or regex) for each
- Define EDM columns for each based on the SIT

E.g.:

29833	→	Street number regex-based SIT
NE 29th St.	→	Street name dictionary-based SIT (requires a match to street number and city or ZIP nearby)
Redmond	→	City name dictionary-based SIT
98052	→	Zip code regex-based SIT

Then create an EDM SIT pattern for "Customer street address" that uses the street name as primary evidence and a combination of the others plus some other PII as secondary (e.g. street name and number plus first name).

Structured process for defining SIT patterns

- Start by identifying all the combinations of evidence you want to detect in a table.
  - Remove all those that have few values allowed (e.g. gender, “marital status”, etc.) since they don’t add value to the detection.
  - Mark all combinations that are of interest (example below is simplified, complex conditions might require more nuanced markings).
- Color code those rows and columns that have a structured, easy to detect format, that can be detected using a SIT.
  - Any combination that involves that row, can use that field as a primary element. You can remove any column that only has marks.
- If there are remaining combinations that don’t involve a marked row or column, choose the row that is simplest to identify via a dictionary or NER and flag them as well.
  - Mark all the combinations involved as covered.
- Continue until there are no combinations not covered by a primary SIT that can be detected via a SIT.

	Last name	**First name**	**National ID**	**Account #**	**Drivers license**	**Street Address**	**Date of birth**
Last name			X	X	X	X	X
First name			X			X	
National ID	X	X		X	X	X	X
Account number	X		X		X	X	X
Drivers license	X		X	X		X	X
Street Address	X	X	X	X	X		
Date of birth	X		X	X	X		

Multi-token additional evidence

- Limitation: by default, EDM matches additional evidence against *individual words* in the text surrounding a primary element match
  - So it can't match "multi-word" terms (e.g. compound last names, street names, phone numbers with spaces, etc.)
  - E.g.: if the street name column for the matched person includes "Calder Cyn Rd." and the text in a document includes "Sarah J. Connor lives in 309 Calder Cyn Rd.", the system will compare the hashes of "lives", "in", "309", "Calder", "Cyn" and "Rd." against the hash of "Calder Cyn Rd." and won't produce a match.
- Solution: assign a SIT to the additional evidence columns when you refer to them in the EDM SIT pattern, just like when you do it for primary evidence.
- How? That's not in the UI!
  - Indeed, but coming soon.
  - Today, you have to edit the XML for the EDM SIT, e.g.:

```
<ExactMatch id="cb664342-1b60-4451-a98f-68b378ce6f62" patternsProximity="300" dataStore="T-800" >
    <Pattern confidenceLevel="75">
        <idMatch matches="SSN" classification="U.S. Social Security Number (SSN)" />
            <Any minMatches="1" maxMatches="3">
                <match matches="Lastname" />
                <match matches="AddressStreet" classification="Street Names Dictionary"/>
                <match matches="Phone" />
            </Any>
    </Pattern>
</ExactMatch>
```

See https://docs.microsoft.com/en-us/microsoft-365/compliance/sit-modify-a-custom-sensitive-information-type-in-powershell for the process.

EDM Special scenarios

Special columns as primary evidence

- Full names
- Last names
- Street addresses
- Account number / Medical Record Numbers with mixed patterns
- Date of birth
- Devices

Full names as primary element

- Don’t use a regular expression, e.g. “([A-Z][a-zA-Z]*){2,3}
  - It will cause lots of f

Special columns as primary evidence

- Full names
- Last names
- Street addresses
- Account number / Medical Record Numbers with mixed patterns
- Date of birth
- Devices

Special columns as primary evidence

- Full names
- Last names
- Street addresses
- Account number / Medical Record Numbers with mixed patterns
- Date of birth
- Devices

Date of birth

Not unique

- If you need to use dates as secondary elements, exclude dates in the future and dates with timestamps:
- `\b([012]?\d|3(0|1))(\/|-|\.)([012]?\d|3(0|1))(\/|-|\.)(19|20)?\d{2}(?! \d{1,2}:\d{2}(:\d{2})?)\b`

The primary element problem

- Named entities (for names, addresses, credentials and other supported entities)
- Dictionaries. This one is non-obvious, but a very flexible solution is to divide the column in question in parts (e.g. a name is divided in first and last names, or an address is divided in number, street name, area/city, zip code, etc.), then collecting all the values in all rows for each of the parts (e.g. all first names, all last names, all street names) and de-duplicate them, then create a keyword dictionary for each list (e.g. a dictionary of all unique first names, one of all unique last names, one for all unique street names, etc.) and using those dictionaries in a SIT that's used for the EDM column in question. While the lists may be large, they are unlikely to exceed the limits of a regular dictionary.
- If a column can't be detected using the techniques above, you can use a regular expression that limits the pattern as much as possible (e.g. for "passwords" you can use a regex that requires the minimum and maximum length and complexity requirements) and include additional requirements so the number of matches is reduced (e.g. a list of keywords typically associated with a password, or some form of account identifier that itself can be detected via a regex such as an email address or an account number nearby).

Gotchas

- Don’t use DoB as a primary field, and be wary of using it even as a secondary field because of formatting variations

Appendix: collateral reading (if you are masochist)

- Sensitive info type definitions: https://aka.ms/sensitiveinfotypes
- Sensitive info type XML syntax for manual edit of SITs: https://docs.microsoft.com/en-us/microsoft-365/compliance/sit-get-started-exact-data-match-create-rule-package
- Configuring EDM: https://docs.microsoft.com/en-us/microsoft-365/compliance/sit-get-started-exact-data-match-based-sits-overview
- Troubleshooting EDM: https://docs.microsoft.com/en-us/microsoft-365/compliance/sit-get-started-exact-data-match-test
- Third party regular expression resources:

  https://regexr.com/ (great tool for learning by trial and error, though it doesn't strictly support the Microsoft syntax)
  http://regexstorm.net/tester (great for troubleshooting, supports the exact Microsoft implementation of regex)
  http://www.rexegg.com/ (extremely thorough regex tutorial)