Support Sensitive Dynamic Properties

Goals

- Allow users to indicate that a dynamic (user-defined) property should be treated as a sensitive property when creating the property.

Background and strategic fit

There have been many asks for users to be able to mark a dynamic property as sensitive.

A good example use case is for a Processor such as InvokeHTTP, which allows arbitrary key-value pairs to be provided and sent as HTTP Headers. A Bearer Token, for instance, may need to be included. This shouldn't be made accessible to just anyone who has access to the flow. It shouldn't be shown in the UI, and its value should be encrypted when written to disk. Currently, there's no way to support this.

Some processors have tried to work around this already, by supporting specific naming patterns, such as "If the property name starts with SENSITIVE." then the property will be treated as sensitive, etc. But this is confusing, not obvious to use, not obvious that it's supported, easy to make a typo that results in the value not being treated as sensitive, etc.

However, it's also important that we not allow dynamic properties of all processors to accept sensitive values. A processor should never be handed a sensitive value unless it explicitly declares that it supports sensitive values, as that is an assertion that it knows how to handle it and won't do things like logging the sensitive value in plaintext, etc.

We should allow processors to declare that they support sensitive dynamic properties by using an annotation.

User interaction and design

When a user chooses to add a new Property to a Processor, Controller Service, or Reporting Task, the UI should expose a new configuration option with a checkbox to indicate whether or not the property is sensitive.

This option should be made available ONLY if the component denotes that it supports sensitive dynamic properties.

If a component is configured with a sensitive value but does not support sensitive values (for example, a processor is changed between Version X and Version Y and in Version Y it no longer supports it), the component should be made invalid and should not have the configured value made available to it.

For any property descriptor that the user declared to be sensitive, its value should be encrypted when the flow is persisted, and its value should not be returned to the UI.

Required Code Changes

This is a large undertaking that will encompass many aspects of the codebase.

It's likely that the entire changeset will be required as a single Pull Request, as it likely can't be broken up in a meaningful way without introducing significant changes that are largely pointless and difficult to test, until the entirety of the feature is complete. The following components will need to be updated:

API

Add new annotation org.apache.nifi.annotation.behavior.SupportsSensitiveDynamicProperties

Framework

ComponentNode
- Update `setProperties(Map<String, String> properties)` and `setProperties(Map<String, String> properties, boolean allowRemovalOfRequiredProperties)` to take a new argument: `Set<String> sensitiveDynamicProperties`
  - This could be a separate method, a `setSensitiveDynamicProperties(Set<String> propertyNames)`, but separating them means that it's easy that one gets updated without the other, and it means that the update is not atomic, so a background thread may treat a sensitive value as non-sensitive.
  - If sensitive dynamic properties are not supported, do NOT throw Exception. Instead, just allow it and component will become invalid. If we throw Exception, this can cause problems later if importing flow into a nifi instance where versions of components are different.

AbstractComponentNode

- Update `getPropertyDescriptor(String propertyName)` to set the sensitivity flag if it's dynamic and name is in the newly added set of sensitiveDynamicProperties
- Update `computeValidationErrors` so that if `!this.sensitiveDynamicProperties.isEmpty() && !this.isSensitiveDynamicPropertiesSupported()` then we add a ValidationResult to `validationResults` and return it immediately.
    - ValidationResult should be invalid
    - Should indicate a sensitive value is configured for dynamic property <abc> but the processor does not support sensitive dynamic properties
    - Should be one of the very first things done in validation. Should return immediately. This ensures that we don't call customValidate() with sensitive values when processor does not support sensitive values, etc. If processor doesn't explicitly state that it allows sensitive values, it should never be given a sensitive value.

StandardProcessContext

- Update to avoid using ProcessorNode.getProcessor().getPropertyDescriptor() and instead use ProcessorNode.getPropertyDescriptor() so that it ensures that the proper sensitivity flag is set

AbstractReportingContext

- Update constructor to take a ReportingTaskNode instead of ReportingTask
- Use ReportingTaskNode.getPropertyDescriptor() instead of ReportingTask.getPropertyDescriptor()

ProcessorResource / ControllerServiceResource / ReportingTaskResource

- Update getPropertyDescriptor() methods to take a query parameter for whether or not it should be reported as sensitive

ProcessorConfigDTO, ControllerServiceDTO, ReportingTaskDTO

- Add a `boolean isSensitiveDynamicPropertySupported`
- Add a `private Set<String> sensitiveDynamicProperties;` and getters/setters

UI

- When creating user-defined property, allow checkbox for whether or not to make sensitive, but only if DTO has TRUE for isSensitiveDynamicPropertySupported
- Include this info in the DTOs going back to server

DtoFactory:

- Update to indicate whether or not component supports sensitive dynamic properties

XmlDocumentationWriter

- Document whether or not user can mark dynamic properties as sensitive

Update anywhere in framework that uses ConfigurableComponent.getPropertyDescriptor

- This shouldn't be used, generally, from the framework because we need to switch to the appropriate ClassLoader anyway

Serialization / Synchronization

// JSON Serialization / synchronization
NiFiRegistryFlowMapper

- Update to populate VersionedProcessor/VersionedCS/VersionedReportingTask with which dynamic properties are sensitive

VersionedFlowSynchronizer

- Update to mark appropriate properties as being sensitive

VersionedProcessGroupSynchronizer

- Update to mark appropriate properties as being sensitive

DifferenceType

- Add new PROPERTY_SENSITIVITY_CHANGED enum value

StandardFlowComparator

- Check for change to sensitivity flag for processor/cs/reporting task

// XML Serialization / synchronization
FlowFromDOMFactory: Check for sensitivity flag enum. If present, use that. If not, check for boolean sensitive flag
StandardFlowSerializer: Write enum for sensitivity flag instead of boolean
FlowConfiguration.xsd: Update to include a new field for sensitivity flag

VersionedPropertyDescriptor

- Deprecate boolean sensitivity methods
- Add enum-based sensitivity methods

VersionedProcessor

- Add Set<String> sensitiveDynamicProperties

VersionedControllerService

- Add Set<String> sensitiveDynamicProperties

VersionedReportingTask

- Add Set<String> sensitiveDynamicProperties

User Guide

Include updated screenshots

Explain that some processors support this, while others don't. Because some processors make no promise to treat values as sensitive

System Tests

Need a new Processor, a new Controller Service, and a new Reporting Task that supports sensitive dynamic properties

Add system test that adds multiple properties, some sensitive, some not. Ensures that when a GET is made to retrieve the values, the ones marked as sensitive do NOT come back (are masked), others do (are not masked). Do this for Processor, Controller Service, AND Reporting Task.
- Also ensure that the property descriptors in the DTO that comes back denote the sensitivity flag appropriately
- Restart, and ensure that the config is still the same after restart.
- Change values. Ensure that values of sensitive and non-sensitive properties updated (may need to write to a file or something some the DTO won't bring back the new values. Ensure that updates occurred. Ensure that DTOs are still correct.

Add test that calls the /nifi-api/processors/{id}/descriptors?name={name}&sensitive={sensitive} to make sure that the returned PropertyDescriptorDTO has accurate value for sensitivity flag.
  - Repeat for Controller Service, Reporting Task

Add test that has a Processor that supports sensitive dynamic properties. Then override getSupportedDynamicProperty. In the property descriptor that is returned, set sensitive(true) to indicate that the property will always be sensitive, regardless of user input. And then ensure that the property value is treated as sensitive (DTO shows descriptor as sensitive and value as masked).

Not Doing

- Support for users to mark arbitrary properties as sensitive or non-sensitive. Only dynamic (user-defined) properties will support allowing user to denote sensitivity.
- Support for dynamic properties to be sensitive for all components. Only components that explicitly indicate that they are willing to handle sensitive values should support being handed sensitive values.