package com.twitter.visibility.rules

import com.twitter.visibility.configapi.params.RuleParams

import com.twitter.visibility.rules.DmConversationRules.\_

import com.twitter.visibility.rules.DmEventRules.\_

import com.twitter.visibility.rules.PolicyLevelRuleParams.ruleParams

object SensitiveMediaSettingsDirectMessagesBaseRules {

val policyRuleParams = Map[Rule, PolicyLevelRuleParams](

NsfwHighPrecisionInterstitialAllUsersTweetLabelRule -> ruleParams(

RuleParams.EnableLegacySensitiveMediaDirectMessagesRulesParam),

GoreAndViolenceHighPrecisionAllUsersTweetLabelRule -> ruleParams(

RuleParams.EnableLegacySensitiveMediaDirectMessagesRulesParam),

NsfwReportedHeuristicsAllUsersTweetLabelRule -> ruleParams(

RuleParams.EnableLegacySensitiveMediaDirectMessagesRulesParam),

GoreAndViolenceReportedHeuristicsAllUsersTweetLabelRule -> ruleParams(

RuleParams.EnableLegacySensitiveMediaDirectMessagesRulesParam),

NsfwCardImageAllUsersTweetLabelRule -> ruleParams(

RuleParams.EnableLegacySensitiveMediaDirectMessagesRulesParam)

)

}

case object DirectMessagesPolicy

extends VisibilityPolicy(

tweetRules = TweetDetailPolicy.tweetRules.diff(LimitedEngagementBaseRules.tweetRules),

dmRules = Seq(

DeactivatedAuthorRule,

ErasedAuthorRule

),

policyRuleParams = SensitiveMediaSettingsDirectMessagesBaseRules.policyRuleParams

)

case object DirectMessagesMutedUsersPolicy

extends VisibilityPolicy(

userRules = Seq(SuspendedAuthorRule)

)

case object DirectMessagesSearchPolicy

extends VisibilityPolicy(

dmConversationRules = Seq(

DropDmConversationWithUndefinedConversationInfoRule,

DropDmConversationWithUndefinedConversationTimelineRule,

DropInaccessibleDmConversationRule,

DropEmptyDmConversationRule,

DropOneToOneDmConversationWithUnavailableParticipantsRule

),

dmEventRules = Seq(

InaccessibleDmEventDropRule,

HiddenAndDeletedDmEventDropRule,

MessageCreateEventWithUnavailableSenderDropRule),

userRules = Seq(ErasedAuthorRule, DeactivatedAuthorRule, SuspendedAuthorRule),

tweetRules =

Seq(ViewerBlocksAuthorRule, ViewerMutesAuthorRule) ++ TweetDetailPolicy.tweetRules.diff(

LimitedEngagementBaseRules.tweetRules),

policyRuleParams = SensitiveMediaSettingsDirectMessagesBaseRules.policyRuleParams

)

case object DirectMessagesPinnedPolicy

extends VisibilityPolicy(

dmConversationRules = Seq(

DropDmConversationWithUndefinedConversationInfoRule,

DropDmConversationWithUndefinedConversationTimelineRule,

DropInaccessibleDmConversationRule,

DropEmptyDmConversationRule,

DropOneToOneDmConversationWithUnavailableParticipantsRule

),

dmEventRules = Seq(

InaccessibleDmEventDropRule,

HiddenAndDeletedDmEventDropRule,

MessageCreateEventWithUnavailableSenderDropRule),

userRules = Seq(ErasedAuthorRule, DeactivatedAuthorRule, SuspendedAuthorRule),

tweetRules =

Seq(ViewerBlocksAuthorRule, ViewerMutesAuthorRule) ++ TweetDetailPolicy.tweetRules.diff(

LimitedEngagementBaseRules.tweetRules),

policyRuleParams = SensitiveMediaSettingsDirectMessagesBaseRules.policyRuleParams

)

case object DirectMessagesConversationListPolicy

extends VisibilityPolicy(

dmConversationRules = Seq(

DropDmConversationWithUndefinedConversationInfoRule,

DropDmConversationWithUndefinedConversationTimelineRule,

DropInaccessibleDmConversationRule,

DropEmptyDmConversationRule,

DropOneToOneDmConversationWithUnavailableParticipantsRule

),

userRules = Seq(ErasedAuthorRule, DeactivatedAuthorRule, SuspendedAuthorRule),

tweetRules =

Seq(ViewerBlocksAuthorRule, ViewerMutesAuthorRule) ++ TweetDetailPolicy.tweetRules.diff(

LimitedEngagementBaseRules.tweetRules),

policyRuleParams = SensitiveMediaSettingsDirectMessagesBaseRules.policyRuleParams

)

case object DirectMessagesConversationTimelinePolicy

extends VisibilityPolicy(

dmEventRules = Seq(

InaccessibleDmEventDropRule,

HiddenAndDeletedDmEventDropRule,

MessageCreateEventWithUnavailableSenderDropRule),

userRules = Seq(ErasedAuthorRule, DeactivatedAuthorRule, SuspendedAuthorRule),

tweetRules =

Seq(ViewerBlocksAuthorRule, ViewerMutesAuthorRule) ++ TweetDetailPolicy.tweetRules.diff(

LimitedEngagementBaseRules.tweetRules),

policyRuleParams = SensitiveMediaSettingsDirectMessagesBaseRules.policyRuleParams

)

case object DirectMessagesInboxPolicy

extends VisibilityPolicy(

dmConversationRules = Seq(

DropDmConversationWithUndefinedConversationInfoRule,

DropDmConversationWithUndefinedConversationTimelineRule,

DropInaccessibleDmConversationRule,

DropEmptyDmConversationRule,

DropOneToOneDmConversationWithUnavailableParticipantsRule

),

dmEventRules = Seq(

InaccessibleDmEventDropRule,

HiddenAndDeletedDmEventDropRule,

DmEventInOneToOneConversationWithUnavailableUserDropRule,

MessageCreateEventWithUnavailableSenderDropRule,

NonPerspectivalDmEventDropRule,

WelcomeMessageCreateEventOnlyVisibleToRecipientDropRule,

GroupEventInOneToOneConversationDropRule

),

userRules = Seq(ErasedAuthorRule, DeactivatedAuthorRule, SuspendedAuthorRule),

tweetRules =

Seq(ViewerBlocksAuthorRule, ViewerMutesAuthorRule) ++ TweetDetailPolicy.tweetRules.diff(

LimitedEngagementBaseRules.tweetRules),

policyRuleParams = SensitiveMediaSettingsDirectMessagesBaseRules.policyRuleParams

)