package com.twitter.visibility.rules

import com.twitter.visibility.configapi.params.FSRuleParams.CardUriRootDomainDenyListParam

import com.twitter.visibility.configapi.params.RuleParam

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

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

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

import com.twitter.visibility.rules.Condition.And

import com.twitter.visibility.rules.Condition.CardUriHasRootDomain

import com.twitter.visibility.rules.Condition.CommunityTweetCommunityVisible

import com.twitter.visibility.rules.Condition.IsPollCard

import com.twitter.visibility.rules.Condition.LoggedOutOrViewerNotFollowingAuthor

import com.twitter.visibility.rules.Condition.Not

import com.twitter.visibility.rules.Condition.Or

import com.twitter.visibility.rules.Condition.ProtectedAuthor

import com.twitter.visibility.rules.Condition.TweetIsCommunityTweet

import com.twitter.visibility.rules.Condition.ViewerIsCommunityMember

object DropProtectedAuthorPollCardRule

extends RuleWithConstantAction(

Drop(Reason.ProtectedAuthor),

And(

IsPollCard,

ProtectedAuthor,

LoggedOutOrViewerNotFollowingAuthor,

)

)

object DropCardUriRootDomainDenylistRule

extends RuleWithConstantAction(

Drop(Reason.Unspecified),

And(CardUriHasRootDomain(CardUriRootDomainDenyListParam))

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableCardUriRootDomainCardDenylistRule)

}

object DropCommunityNonMemberPollCardRule

extends RuleWithConstantAction(

Drop(Reason.CommunityNotAMember),

And(

IsPollCard,

TweetIsCommunityTweet,

Or(

Not(ViewerIsCommunityMember),

Not(CommunityTweetCommunityVisible),

)

),

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableCommunityNonMemberPollCardRule)

override def enableFailClosed: Seq[RuleParam[Boolean]] = Seq(

EnableCommunityNonMemberPollCardRuleFailClosed)

}