contract IDS\_SignatureStorage {

// Structure to store Metadata

struct Metadata {

string signatureID;

string name;

string severityLevel;

string category;

uint16 revisionNumber;

}

// Structure to store Description

struct Description {

string summary;

string targetSystems;

}

// Structure to store Conditions

struct Conditions {

string patternOrRule;

string payloadContent;

string headerInformation;

string anomalousBehavior;

string timing;

string contextualInformation;

}

// Structure to store Actions

struct Actions {

string alert;

string response;

string logging;

}

// Structure to store Examples

struct Examples {

string exampleOfMatchingTraffic;

string falsePositiveConsiderations;

}

// Structure to store References

struct References {

string documentation;

string cveIDs;

}

// Structure to store Additional Information

struct AdditionalInformation {

string attackVector;

string mitigationStrategies;

}

// Structure to store the complete IDS Signature

struct IDSSignature {

Metadata metadata;

Description description;

Conditions conditions;

Actions actions;

Examples examples;

References references;

AdditionalInformation additionalInfo;

}

// Array to store all IDS Signatures

IDSSignature [] public signatures;

// Event to log the storage of a new signature

event SignatureStored(

string signatureID,

string name,

string severityLevel,

string category,

uint16 revisionNumber

);

// Function to store a new IDS Signature

function storeSignature(

string memory \_signatureID,

string memory \_name,

string memory \_severityLevel,

string memory \_category,

uint16 \_revisionNumber,

string memory \_summary,

string memory \_targetSystems,

string memory \_patternOrRule,

string memory \_payloadContent,

string memory \_headerInformation,

string memory \_anomalousBehavior,

string memory \_timing,

string memory \_contextualInformation,

string memory \_alert,

string memory \_response,

string memory \_logging,

string memory \_exampleOfMatchingTraffic,

string memory \_falsePositiveConsiderations,

string memory \_documentation,

string memory \_cveIDs,

string memory \_attackVector,

string memory \_mitigationStrategies

)

public IDSSignature memory newSignature = IDSSignature({

metadata: Metadata( signatureID: \_signatureID, name:\_name, severityLevel: \_severityLevel,

category: \_category, revisionNumber: \_revisionNumber

}),

description: Description( summary: \_summary, targetSystems: \_targetSystems),

conditions: Conditions({ patternOrRule: \_patternOrRule, payloadContent: \_payloadContent,

headerInformation: \_headerInformation, anomalousBehavior: \_anomalousBehavior,

timing: \_timing, contextualInformation: \_contextualInformation),

actions: Actions( alert: \_alert, response: \_response, logging: \_logging),

examples: Examples({ exampleOfMatchingTraffic: \_exampleOfMatchingTraffic,

falsePositiveConsiderations: \_falsePositiveConsiderations }),

references: References( documentation: \_documentation, cveIDs: \_cveIDs }),

additionalInfo: AdditionalInformation({ attackVector: \_attackVector, mitigationStrategies:

\_mitigationStrategies}) );

signatures.push(newSignature);

emit SignatureStored(\_signatureID, \_name, \_severityLevel, \_category, \_revisionNumber);

}

// Function to retrieve all stored IDS Signatures

function getAllSignatures() public view returns (IDSSignature[] memory) {

return signatures;

}

}