openXSAM - Towards a common Security Analysis Exchange Format for ISO/SAE 21434 and UN Reg. 155

Wouldn't it be great if you could export cybersecurity data from your TARA tool and import it into your requirement management platform or verification environment and vice versa? One of the biggest challenges in automotive security is building cohesion between tools to speed up cybersecurity engineering. https://openxsam.io seeks to achieve this by building an open format to exchange security information for vehicles.

openXSAM stands for **open** format for e**X**changing **S**ecurity **A**nalysis **M**odels.
This industry paper describes why it has been created, what the current considerations are, and where we would like to head next.

Management Summary

New regulations and norms like ISO21434 and UNECE require the automotive industry to perform and document security risk analysis activities and results. This is true for the development process as well as for the life cycle of the products. As a result, it becomes important for the parties to integrate security risk analysis software in existing tool chains. The parties would also benefit from an exchange format that would allow the exchange of security risk analysis data across departments and corporations.
openXSAM could serve as a protocol to achieve the above goals.
This paper outlines the status quo of the format and describes some of the use cases.
It serves as a basis for joint future work.
The work on openXSAM will be open to all parties interested in establishing an open exchange format for security risk analysis in the automotive domain.

Some aspects of openXSAM have been developed as part of the R&D project SecForCARs (https://www.forschung-it-sicherheit-kommunikationssysteme.de/projekte/sicherheit-fuer-vernetzte-autonome-fahrzeuge) in collaboration with partners.

Why openXSAM?

Ensuring that modern vehicles are safe and secure has become an important part of vehicle design and manufacturing and their associated processes.
ISO/SAE 21434 and UN Reg. 155 now give a normative framework for security related activities. These standards focus on WHAT needs to be done. They define basic guidelines

regarding the risk analysis process and the analysis methods. They also provide information regarding required work products including some examples (e.g. report tables). The standards expect OEMs and their suppliers to ensure an efficient life-cycle-management, but fall short to provide clear instructions on HOW to achieve this.

Status Quo of the automotive development process

- Highly distributed (OEM, Tier 1, Tier 2, …)
- Similar high level structure of development process (V-model)
- Model based engineering approach
- Various heterogeneous software tools are used within tool chains
- Tracing of related artifacts is required across the tool chain

Security risk analysis is mandatory for many actors in the automotive industry. This includes all OEMs and suppliers. Similar to safety, security aspects require integration across the entire V-model: Design, specification, development, testing, monitoring of the product in the field, and incident management.
The input and the output of the security risk analysis process each require tool chain integration for validation, verification, and tracing purposes.
This includes the following artifacts and work products

- Item definition (e.g. involved technologies and their version, functionality, components, connections, data, data flows etc.)
- Assumptions and analysis scope
- Cybersecurity Risks, treatment decisions and Controls / countermeasures
- Cybersec. concept with requirements and security claims
- Monitoring and incident response

Use cases for an open exchange format for model based security risk analysis

1. Standardized interfaces for tool chain integration of security risk analysis solutions
   - Import of required security risk analysis parameters (requirements, functions, architecture etc.)
   - Export of security risk analysis results (risks, recommended controls and related risk levels)
   - Tracing of relations and artifacts across the development process. All security artifacts are subject to change management, requirements management and document management. In these existing platforms, the security artifacts need to be stored in an actionable format, so that later parts of the chain may start from there.
   - Security tools build on each others' work products. For example, a threat modeling tool, an intelligence database and a risk management tool need information from each other.

2. Standardized exchange format for security risk analysis artifacts across organizations
   - Exchange of security analysis data between departments of larger organizations

- Exchange across company boundaries in value chains (Tier 2 -> Tier 1 -> OEMs)
- Handover of security entities as contractual deliverables
- The diversity of specialization across organizations and teams leads to a diversity of security tools (think penetration-testing vs. security by design).
- The diversity of tools requires an interchange format so that they chain up without manual copying.

3. Integration point for solutions and databases that hold security entities

The integration needs to include solutions for new enterprise workflows. There are new workflows to perform continuous cybersecurity activities, such as monitoring and incident response. The solutions to support these activities are just growing. For instance, an incident database for responsible disclosure will commonly trigger updating a risk assessment model. This could lead to implementation of controls in the system tool and a reassessment of safety and security requirements. These new non-V workflows require integration with security databases to stay manageable.

When security workflows cross organizational borders, there may be multiple databases that need to responsibly be kept in sync. There is an open question on how to collaborate on incidents across organizational borders. Once the question on WHAT to communicate is answered, an interchange format will be required to provide the HOW for that.

Status of XSAM Development

The XSAM format has so far evolved as part of the research project SecForCARs (https://www.forschung-it-sicherheit-kommunikationssysteme.de/projekte/sicherheit-fuer-vernetzte-autonome-fahrzeuge), and a few additional collaborations on the side of itemis. We now aim at carrying it outside itemis and creating a shared-ownership format openXSAM together with additional parties.
The existing XSAM format is already supported by itemis SECURE and in productive use for specific use cases. As part of this initiative, we expect the format to evolve, as it generalizes to more use cases and grows more mature.

The openXSAM Format allows to express the following data in line with ISO/SAE 21434

- Parameters describing the applied security risk analysis method
    - Impact categories
    - Impact level
    - Attack feasibility model (such as the available factor levels in case of a attack-potential based method, or CVSS parameters)
    - Mapping of impact levels and feasibility levels to risk levels

- Description of the items/ assets in a security-relevant format
    - Functions
    - Components
    - Connections
    - Data
    - Data flows

- Security Model Elements
    - Damage scenarios and security objectives
    - Threat scenarios
    - Attack trees
    - Applied controls
    - Resulting risks and their treatment decisions

- Catalog database
    - Deduplicate and align feasibility ratings for common threats
    - Have a solutions catalog that contains controls for given common threats
    - Technologies and versions as a traceability joint point (e.g. for gray-box vulnerability testing or incident response)
- Cybersecurity concept
    - Goals, requirements, related R155 mitigations

Technical Design Considerations

- Allow optional external IDs for all elements to build a traceability solution that knows dependencies down to the level of single elements (such as threats), respecting organization-local flavors
- Hold external IDs to allow repeated import
- Allow holding no IDs to allow tiny constructive integrations without any ID knowledge
- Defer resolution of references to allow merge points of two artifacts in the tool chain (such as an assessment model from CC and an analysis from a supplier)
- Allow reference by name, internal ID and external ID to build integrations of varying ownership and knowledge of tracing IDs. For example, an attack rating of the knowledge factor level “EXPERT” would rather be name-based, because it is easier

to build integrations with a readable format, and also because it is resilient over assessment-model changes.
- Be prepared for dialects based on xml namespacing (as we can expect organization specific customizations)
- Optional model IDs to have control over updating multiple analysis models (for example when syncing databases) or copying from one model to another (for instance to fork from one analysis to start analysis a variant of the component)
- Reference other elements in descriptions so that semi-structured text may link to related security entities (it is a markdown-ish syntax when using the name)
- Names for unique yet readable references, titles for self-explanatory identifiers and descriptions to hold additional information on request

Model for joint work on openXSAM

We seek out to form a circle of partners to jointly work on such an open exchange format for security analyses.

These are current tasks for joint work we identified:
- Documentation of status quo of openXSAM
- Identification of use cases

Results of our joint work could look like follows:
- Fully documented openXSAM format including XML-schema
- Documentation of use cases for tool chain integration
    - Imports to security analysis tools via openXSAM
    - Exports from security analysis tools via openXSAM
    - Tracing across tool chain via openXSAM
- Availability of sample projects / data for the defined use cases
- Open access to online resources for documentation and use cases (www.openxsam.io)
- Commitment to support openXSAM from security analysis tool vendors
- Mid-term: international standardization of format

We aim to build this up as an open community across vendors, countries and organizations. If you read this and would like to participate, you are welcome to approach us at leopold@itemis.de .

Use Cases

Example 1: Import System Architecture

It is a common approach to base a security analysis on existing system architecture. That system could come from tagging the security-relevant elements in the system modeling tool and then importing those marked system elements into the threat modeling tool.

Let's look at the openXSAM code that exchanges the description of a component and a function:

```
<Component name="BCU" title="Body Control Unit" description=""
ea:id="6WGPIfYZR_A">
   <Technologies/>
   <StoredData/>
   <SubComponents/>
   <AssignedFunctions>
       <FunctionAssignment target="ea/6WGPIfYZRzm"/>
   </AssignedFunctions>
   <Todos>
       <Todo done="false" group="TODO" text="Review for security-relevant
details"/>
   </Todos>
</Component>
<Function name="OnFunction" title="Switch Headlamp On" description=""
ea:id="6WGPIfYZRzm">
   <SubFunctions/>
   <Todos/>
</Function>
```

This figure lets us note a few things:

- The name and title of the two is taken from the system modeling tool
- The elements are annotated and referenced with their ids which they received in the system modeling tool. When importing this into the threat modeling tool, the existing ids can be remembered for traceability and used to match the elements again down the stream
- This importer is annotating our component with the todo "Review for security-relevant details". Since the system models of itemis SECURE tend to diverge from the designed system model (e.g. describing the encryption algorithm parameters), some more refinement work will usually be needed here.
- Starting with this import provides the benefit that elements are already linked to the general system architecture. This link also allows reasoning about coverage and dependencies down the stream.

Example 2: Transfer Security Objectives

Let's imagine that our organization uses two separate tools for modeling security objectives and assessing risks. openXSAM is used to transfer the security objectives from one tool to the other tool so that the risk assessment can take them into account without manually copying any values.

```
<SecurityObjective name="G.3" title="Integrity Server Response"
    ysa:id="2gFkVNfFR1y"
    threatenedBy="or(MitM, and(AdmPC, TampSServ))">
   <InstantiatedClasses>
      <SecurityPropertyRef ysa:id="2gFkVNf" target="name/SecurityProperty/INT"/>
   </InstantiatedClasses>
   <ConcernedTOEEs>
      <TOEERef ysa:id="2gFkVNfFR1$" target="ysa/1E_VH$V8u63"/>
   </ConcernedTOEEs>
   <ExplicitDP/>
   <DamageScenarios>
      <DamageScenarioRef damageScenario="ysa/7sK5zqfx1Ci" ysa:id="7sK5zqfx1Co"/>
   </DamageScenarios>
   <Todos/>
</SecurityObjective>
```

This is an example XSAM of a Security Objective. Let's walk through some details of it:

- It has the name "G.3" and the title describes it to be about the integrity of a server response.
- It comes with an internal ID that is provided from the tool itemis SECURE, which traceability tools can rely on to track it. There may be IDs for multiple tools provided, since the id of another system for the same entity could also be known.
- The threatenedBy is an expression language that is used to describe the dependency between security entities. *Or* is used to describe the independence of the referenced elements, while *And* claims dependence between the referenced elements. In this example, the Integrity of the server response could be broken by a man in the middle attack (MitM) or by doing both, compromising a system administrator's machine (AdmPC) and tampering with the server (TampServ).
- The security objective references the security property *INT* by its name. The reference itself has an id as well to ease merging incomplete information (it allows to tell apart changing a reference from adding one and not knowing about the other)
- This security objective has a reference to a particular target of evaluation element (TOEE). In this case, the target TOEE is a data element, called Server Response (not visible in the XML).
- The security objective also has a reference to a damage scenario, which rates the potential impact that could yield if this objective was violated. In our example, the target damage scenario has rated a life threatening impact and an unusable vehicle as potential damage (not visible in the XML).
- No ExplicitDP (explicit damage potential) and no Todos are modeled at this security objective right now

Example 3: An update from an intelligence database

In this use case an *Intelligence Database* informs on updated Attack Feasibility Ratings based on recent 0-days. This rating updates the required time for a *man in the middle attack* down to *HOURS*. It also enforces the description, name, title to have the expected values. Other properties of the threat are left untouched.

The example demonstrates how three features allow formulating small, low-overhead XSAM files. First, absent collections will not be modified. Second, collections with `<ExistingElements />` will only update, but not delete. Third, name-references can be used to point to the assessment model so that cryptic IDs do not distract from reading the XSAM document, such as in `riskFactor="name/Feasibility Category/TIME"`.

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurityChunk
   xmlns="http://www.openxsam.io/xmlns/xsam"
   xmlns:ysa="http://www.itemis.de/yakindu-security-analyst" ysa:id="1E_VH$V8tTo"
ysa:modelRef="r:2b50e850-9e01-4842-8a19-0558d10cdd14(xsam@tests)"
   name="SecurityChunk">
   <Elements>
       <!-- in the description: link the words to the security properties AUT and INT
-->
       <!-- in the id: provide an identifier to find the updated threat -->
       <Threat description="Exploit client/server communication channel
[authentication|AUT] and [integrity|INT] "
           id="ysa/r:2b50e850-9e01-4842-8a19-0558d10cdd14(xsam@tests)/1UEFqBLBtFT"
           name="MitM" title="Manipulation and information disclosure">
           <InitialRiskFactors>
               <!-- let's reference these by name to ease building the exporter -->
               <RiskFactorRating riskFactor="name/FeasibilityCategory/TIME"
level="name/FeasibilityOption/HOURS" />
           </InitialRiskFactors>

           <!-- unlisted property elements of this threat are not changed -->
           <!-- since we're not providing any <Todos/> element, possibly present todos
are kept intact -->
           <ExistingElements />
       </Threat>

       <!-- existing elements in this chunk will be left untouched -->
       <ExistingElements />
   </Elements>
</SecurityChunk>
```