AutomationML Security Extension for Cyber-Physical Systems

1 November 2025

1 Introduction

This specification describes an extension of the Automation Markup Language (AutomationML) [1] for cyber-physical systems (CPS), based on the AMLSec framework [2]. It includes customised representations of assets, vulnerabilities, hazards, and relationships that are specifically relevant to CPS security modelling.

The specification is intended to be read in conjunction with the official AutomationML standard, formalised as IEC 62714 [4], as fundamental elements such as `InternalElement` and related constructs may be unfamiliar to some readers. Building on the foundational definitions provided by AutomationML, IEC 62714, and AMLSec, this document extends these standards to support quantitative metrics – including the Common Vulnerability Scoring System (CVSS) [3] and the Exploit Prediction Scoring System (EPSS) [5] – as well as mitigation tracking, thereby facilitating advanced risk assessments and engineering applications.

2 Core Elements

2.1 Assets

- `InternalElement` object
- Attributes:
  - `Vendor`: String — Manufacturer name
  - `Version`: String — Asset version
  - `FailureRatePerHour`: Numeric — Estimated based on published literature and technical specifications of comparable components available online.
  - `Impact Rating`: Numeric — Captures the multifaceted consequences of successful attacks across safety, operational, financial, and informational domains.
  - `Date of first use`: String — Date representing the start of asset utilisation.
- Interfaces: `ExternalInterface` for system/logical links
- Class path: `AssetOfICS/SoftwareApplication`

2.2 Vulnerabilities

- `InternalElement` object
- Attributes:
  - `CVE`: String — Unique Common Vulnerabilities and Exposures identifier for a publicly known vulnerability.
  - `CVSS`: String — Full Common Vulnerability Scoring System vector string.
  - `Attack Name`: String — Description of the attack.
  - `Probability of Impact`: Numeric — Probability that the vulnerability will have a severe impact if exploited.
  - `Probability of Exposure`: Numeric — Likelihood that the vulnerability can be exposed to attacks.
  - `Probability of Mitigation`: Numeric — Likelihood that a mitigation measure is effective.
  - `EPSS`: Numeric — Exploit Prediction Scoring System value representing probability of exploit (optional: for vulnerabilities not linked to CVEs).
- Interfaces: `ExternalInterface` to assets
- Class path: `VulnerabilityforSystem/Vulnerability`

2.3 Hazards

- `InternalElement` object
- Attributes:
  - `Impact Rating`: Numeric — Severity rating of the hazard.
  - Other attributes: `Consequence`, `Causes` — Additional hazard-specific attributes.
- Interfaces: `ExternalInterface` to assets/components
- Class path: `HazardforSystem/Hazard`

2.4 Relationships

- `InternalLink` object
- Attributes:
  - `Name`: Identifier
  - `RefPartnerSideA`: Source ID
  - `RefPartnerSideB`: Target ID

3 Relationship Rules

To ensure consistency and correctness in modelling cyber-physical system security, the following relationship rules apply to nodes representing Assets, Vulnerabilities, Hazards, and Goals within AutomationML:

- **Asset nodes** may have children that are:
  - Asset nodes
  - Vulnerability nodes
  - Hazard nodes
- **Hazard nodes** may have children that are:
  - Asset nodes
  - Hazard nodes
- **Vulnerability nodes** may have children that are:
  - Asset nodes
  - Vulnerability nodes
- **Asset, Vulnerability, or Hazard nodes** may have children that are:
  - Goal nodes (special case)
- **Goal nodes** represent ultimate objectives (modeled as Hazards) and are leaf nodes without children.

These rules enforce valid hierarchical and semantic relationships within the security modelling framework, reflecting realistic attack paths and system dependencies.

4 Class Libraries and Attribute Definitions

4.1 Interface Class Library: ConnectionBetnAssets

Defines standard interface types for linking assets, vulnerabilities, hazards, and users.

- `Network`: For network-level connections.
- `Logic`: For logical or protocol-level interfaces.
- `User-Interaction`: For user-related interfaces.

Example snippet:

```
<InterfaceClassLib Name="ConnectionBetnAssets">
  <InterfaceClass Name="Network" />
  <InterfaceClass Name="Logic" />
  <InterfaceClass Name="User-Interaction" />
</InterfaceClassLib>
```

4.2 Role Class Library: Requirements

Defines roles relevant for system components and processes.

- `Process`
- `Safety`
- `Security`
- `Communication`

Example snippet:

```
<RoleClassLib Name="Requirements">
  <RoleClass Name="Process" />
  <RoleClass Name="Safety" />
  <RoleClass Name="Security" />
  <RoleClass Name="Communication" />
</RoleClassLib>
```

4.3 System Unit Class Library: AssetOfICS

Models the hierarchical classification of assets including hardware, software, and user elements, describing their components and interfaces.

Key sub-elements include:

- Hardware: Process devices (Sensors, Actuators, Controllers, Workstations, Servers), Machines, Network Devices (Switch, Router, Gateway, Firewall).
- Software: Firmware/Operating system, Applications, Process logic, OT adapters.
- User with user-related interface.

Example snippet:

```
<SystemUnitClassLib Name="AssetOfICS">
  <SystemUnitClass Name="Hardware">
    <SystemUnitClass Name="Process device">
      <SystemUnitClass Name="Sensor">
        <ExternalInterface Name="SensorOP" RefBaseClassPath="
            ConnectionBetnAssets/Network based" />
        <ExternalInterface Name="SensorOP" RefBaseClassPath="
            ConnectionBetnAssets/Logic based" />
      </SystemUnitClass>
      <!-- Other system unit classes omitted for brevity -->
    </SystemUnitClass>
  </SystemUnitClassLib>
```

4.4 System Unit Class Libraries for Hazards and Vulnerabilities

- `HazardforSystem`: Defines hazard entities with interface references.
- `VulnerabilityforSystem`: Defines vulnerability entities including key attributes like CVE and CVSS, and interfaces.

Example snippet:

```
<SystemUnitClassLib Name="HazardforSystem">
  <SystemUnitClass Name="Hazard">
    <ExternalInterface Name="HazardRef" RefBaseClassPath="
        ConnectionBetnAssets/HazardRef" />
  </SystemUnitClass>
```

```
</SystemUnitClassLib>

<SystemUnitClassLib Name="VulnerabilityforSystem">
  <SystemUnitClass Name="Vulnerability">
    <Attribute Name="CVE" AttributeDataType="xs:string" />
    <Attribute Name="CVSS" AttributeDataType="xs:string" />
    <Attribute Name="Attack Name" AttributeDataType="xs:string"
        />
    <ExternalInterface Name="VulnerabilityRef" RefBaseClassPath=
        "ConnectionBetnAssets/VulnerabilityRef" />
  </SystemUnitClass>
</SystemUnitClassLib>
```

4.5 Attribute Type Library

Defines reusable attribute structures for equipment, hazards, and vulnerabilities.

- `AutomationEquipments`: Vendor, Part, Product, Version, FailureRatePerHour, Date of first use.
- `Hazard`: Name, Severity, Probability, Consequence, Causes.
- `Vulnerability`: CVE, CVSS, Attack Name, Probability of Impact, Probability of Mitigation, Probability of Exposure.

Example snippet:

```
<AttributeTypeLib Name="AttributeTypeLib">
  <AttributeType Name="AutomationEquipments">
    <Attribute Name="Vendor" AttributeDataType="xs:string" />
    <Attribute Name="Version" AttributeDataType="xs:string" />
    <Attribute Name="FailureRatePerHour" AttributeDataType="xs:
        float" />
    <Attribute Name="Date of first use" AttributeDataType="xs:
        string" />
  </AttributeType>
  <AttributeType Name="Hazard">
    <Attribute Name="Severity" AttributeDataType="xs:string" />
    <Attribute Name="Probability" AttributeDataType="xs:string"
        />
    <Attribute Name="Consequence" AttributeDataType="xs:string"
        />
    <Attribute Name="Causes" AttributeDataType="xs:string" />
  </AttributeType>
  <AttributeType Name="Vulnerability">
    <Attribute Name="CVE" AttributeDataType="xs:string" />
    <Attribute Name="CVSS" AttributeDataType="xs:string" />
    <Attribute Name="Attack Name" AttributeDataType="xs:string"
       />
    <Attribute Name="Probability of Impact" AttributeDataType="
       xs:string" />
    <Attribute Name="Probability of Mitigation"
       AttributeDataType="xs:string" />
    <Attribute Name="Probability of Exposure" AttributeDataType=
       "xs:string" />
  </AttributeType>
</AttributeTypeLib>
```

5 XML Schema Structure Examples

5.1 Asset Example

```
<InternalElement Name="MobileWeb Application" ID="MobileWebApp"
    RefBaseSystemUnitPath="AssetOfICS/SoftwareApplication">
  <Attribute Name="Vendor" AttributeDataType="xs:string" Value="
     AcmeCorp"/>
  <Attribute Name="Version" AttributeDataType="xs:string" Value=
     "1.2.3"/>
  <Attribute Name="FailureRatePerHour" AttributeDataType="xs:
     float" Value="0.09"/>
  <Attribute Name="Impact Rating" AttributeDataType="xs:float"
     Value="0.09"/>
  <Attribute Name="Date of first use" AttributeDataType="xs:
     string" Value="2024-01-01"/>
  <ExternalInterface Name="ToVulnerability" ID="AppToVuln"
     RefBaseClassPath="ConnectionBetnAssets/VulnerabilityRef"/>
</InternalElement>
```

5.2 Vulnerability Example

```
<InternalElement Name="V1 CVE-2024-50684" ID="V1"
    RefBaseSystemUnitPath="VulnerabilityforSystem/Vulnerability">
  <Attribute Name="CVE" AttributeDataType="xs:string" Value="CVE
     -2024-50684"/>
  <Attribute Name="CVSS" AttributeDataType="xs:string" Value="
     CVSS3.1/AV:N/AC:H/..."/>
  <Attribute Name="Attack Name" AttributeDataType="xs:string"
     Value="Remote Injection"/>
  <Attribute Name="Probability of Impact" AttributeDataType="xs:
     float" Value="0.65"/>
  <Attribute Name="Probability of Exposure" AttributeDataType="
     xs:float" Value="0.00049"/>
  <Attribute Name="Probability of Mitigation" AttributeDataType=
     "xs:float" Value="0.90"/>
  <Attribute Name="EPSS" AttributeDataType="xs:float" Value="
     0.95"/>
  <ExternalInterface Name="ToAsset" ID="VulnToApp"
     RefBaseClassPath="ConnectionBetnAssets/VulnerabilityRef"/>
</InternalElement>
```

5.3 Hazard Example

```
<InternalElement Name="H1 Data Leakage" ID="H1DataLeakage"
   RefBaseSystemUnitPath="HazardforSystem/Hazard">
  <Attribute Name="Impact Rating" AttributeDataType="xs:float"
     Value="0.13"/>
  <ExternalInterface Name="ToAsset" ID="HazardToApp"
     RefBaseClassPath="ConnectionBetnAssets/HazardRef"/>
</InternalElement>
```

5.4 Relationship Example

```
<InternalLink Name="MobileWebAppToV1" RefPartnerSideA="
   MobileWebApp" RefPartnerSideB="V1"/>
```

6 Specification Compliance and Extensibility

This specification adheres to the IEC 62714 (AutomationML/CAEX) standard, leveraging standard class and attribute type libraries to represent cyber-physical system assets, hazards, and vulnerabilities with semantic clarity. Security-specific attributes and relationships are modelled using extended attributes`<Attribute Name="..." ... />` and `ExternalInterface`/`InternalLink` constructs referencing elements by ID [1, 6].

References

[1] AutomationML Consortium. Specifications – AutomationML, 2025. URL: https://www.automationml.org/about-automationml/specifications/.

[2] Matthias Eckhart, Andreas Ekelhart, and Edgar Weippl. Automated security risk identification using AutomationML-based engineering data. *IEEE Transactions on Dependable and Secure Computing*, 19(3):1655–1672, 2020.

[3] FIRST.Org, Inc. Common Vulnerability Scoring System v3.1: Specification Document, 2019. Version 3.1. Available at https://www.first.org/cvss/v3-1/specification-document. URL: https://www.first.org/cvss/v3-1/specification-document.

[4] International Electrotechnical Commission. Engineering data exchange format for use in industrial automation systems engineering - Automation Markup Language - Part 1: Architecture and general requirements, April 2018. Redline version; includes changes from the previous edition highlighted in track changes mode. URL: https://webstore.iec.ch/en/publication/32339.

[5] Jay Jacobs, Sasha Romanosky, Benjamin Edwards, Michael Roytman, and Idris Adjerid. Exploit Prediction Scoring System (EPSS). https://www.first.org/epss/, 2021. EPSS project maintained by FIRST.Org, Inc. Version 4 released March 2025.

[6] OPC Foundation. General Information to AutomationML and OPC UA, 2011. URL: https://reference.opcfoundation.org/AML/v100/docs/4.