GUIDELINE

Axis security development model

February 2022

Table of contents

1. Introduction

1.1 ASDM objectives

The Axis security development model (ASDM) is a framework that defines the process and tools used by Axis to build software with security built-in throughout the lifecycle, from inception to decommission.

The primary objectives driving ASDM efforts are:

> Make software security an integrated part of Axis software development activities
> Reduce security related business risks for Axis customers
> Meet increasing awareness of security considerations by customers and partners
> Create potential for cost reduction because of early detection and resolution of issues

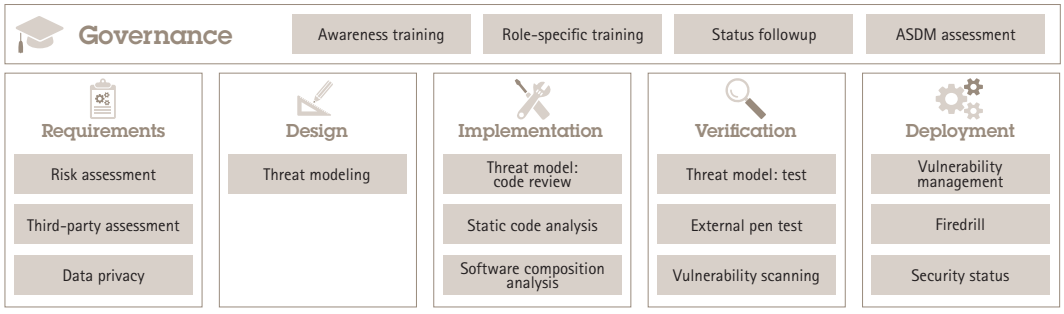
ASDM scope is all Axis software included in Axis products and solutions. The Software Security Group (SSG) is owner and maintainer of the ASDM.

1.2 Glossary

ASDM	Axis security development model
SSG	Software Security Group
Firmware Steering Group	R&D management
Satellite	Developers who have a natural affinity for software security
Vulnerability board	Axis contact point in relation to vulnerabilities found by external researchers
Bug bar	Security target for a product or solution
DFD	Data flow diagram

2. ASDM overview

The ASDM comprises several activities spread across the major development phases. The security activities are collectively identified as the *ASDM*.



The SSG is responsible for governing the ASDM and evolving the toolbox over time. There is an ASDM roadmap and a rollout plan for implementing new activities and increasing ASDM maturity across the development organization. Both the roadmap and roll-out plan are owned by the SSG, but the responsibility for actual implementation in practice (i.e., performing activities related to development phases) is delegated to the R&D teams.

2.1 Software Security Group (SSG)

SSG is the main internal contact entity towards development organizations for security related issues. It is comprised of Security Leads and others with specialist security knowledge in development areas such as requirements, design, implementation, verification, as well as cross-functional DevOps processes.

SSG is responsible for development and maintenance of the ASDM for secure development practices and security awareness in the development organization.

2.2 Satellites

Satellites are members of the development organization that spend a part of their time working with software security aspects. The reasons for having satellites are:

> Scale ASDM without building a large central SSG
> Provide ASDM support close to the development teams
> Facilitate knowledge sharing, e.g., best practices

A satellite will assist in implementing new activities and maintaining the ASDM in a subset of the development teams.

2.3 ASDM activity roll-out

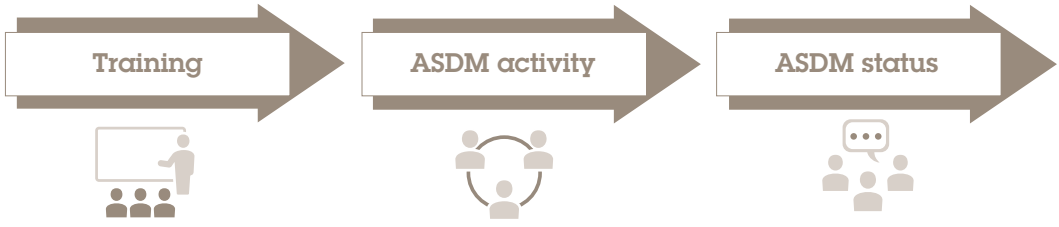
ASDM activity roll-out to a development team is a staged process:

1. The team is introduced to the new activity through role-specific training.
2. SSG works together with the team to perform the activity, e.g., risk assessment or threat modeling, for selected parts of the system(s) managed by the team.
3. Further activities related to integrating the toolbox in daily work will be handed over to the team and satellite when they are ready to work independently without direct SSG involvement. In this phase, the work is governed by the team manager through the ASDM status.

The rollout is repeated when there are new versions of the ASDM available with modified and/or added activities. The amount of time spent by SSG with a team is highly dependent on the activity and code complexity. But it may range from 2-3 calendar weeks of work performed in 4-6 workshops

A key factor for successful handover to the team is the existence of an embedded satellite who can continue further ASDM work with the team. SSG drives learning and assignment of the satellite in parallel with activity rollout.

The figure below summarizes the rollout methodology.



SSG definition of "done" for handover is:

- Role specific training performed
- Satellite assigned
- Team is ready to perform the ASDM activity
- Recurring ASDM status meetings established

SSG use input from the teams to assemble status reports towards senior management.

2.4 Other SSG activities

In parallel with roll-out activities, the SSG conducts broader security awareness training activities targeting e.g., new employees and senior management. Additionally, SSG maintains a security heat map of Axis solutions for overall/architectural risk assessment purposes. Proactive security analysis activities for specific modules are performed based on the heat map.

2.5 Roles and responsibilities

As shown in in the table below, there are some key entities and roles which are part of the ASDM program. The table below summarizes roles and responsibilities in relation to the ASDM.

Role/Entity	Part of	Responsibility	Comment
Security expert	SSG	Govern ASDM, evolve the toolbox and drive ASDM rollout	100% assigned to SSG
Satellite	Development line	Help SSG to implement ASDM the first time, coach teams, perform trainings and ensure that the team can continue using the Toolbox as part of the daily work, independently from SSG. Cross-team responsibility (several teams) required to constrain total number of Satellites.	Interested and engaged developers, architects, managers, testers, and similar roles who have a natural affinity for software security. Satellites assign at least 20% of their time to ASDM related work.
Managers	Development line	Secure resources for implementation of ASDM practices. Drive tracking and reporting on ASDM status and coverage.	Development teams own ASDM implementation, with SSG as a support resource.
Firmware Steering Group (FW SG)	R&D management	Decides on security strategy and acts as main SSG reporting channel.	SSG reports to FW SG on a regular basis.

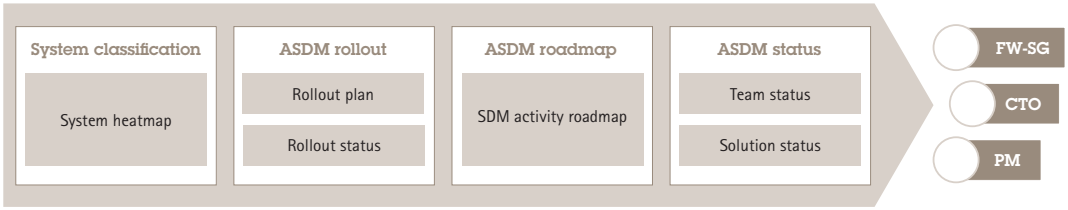
3. ASDM governance

The governance system comprises the following parts:

> System risk heatmap to help prioritize ASDM activities
> Rollout plan and status to focus training efforts
> Roadmap to evolve the toolbox
> Status to measure how well the ASDM activities are integrated in the organization

The ASDM system is thus supported from both a tactical/operational perspective as well as from a strategic/ executive perspective.

Executive guidance on the right-hand side in the figure has a focus on how to develop the organization for optimal effectiveness in line with Axis business goals. An important input to this is the ASDM status reporting performed by SSG towards the Firmware Steering Group, CTO and Product Management.



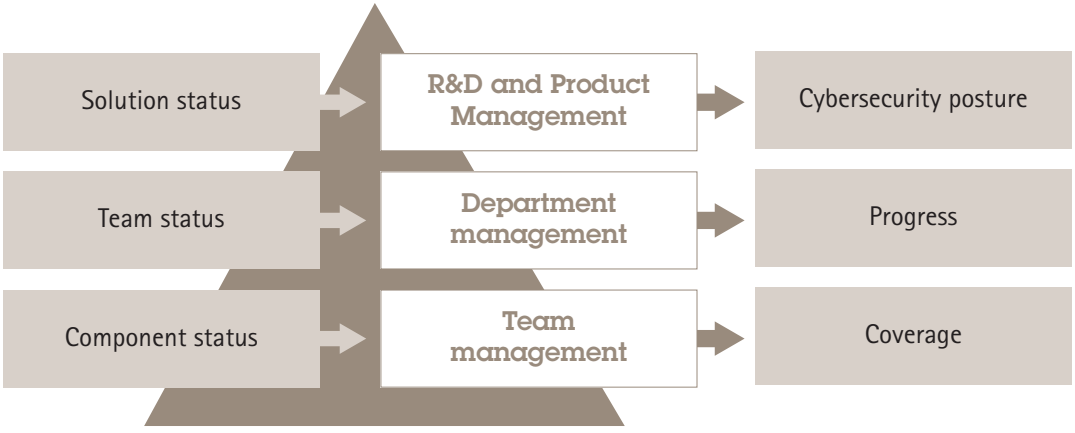
3.1 ASDM status structure

The ASDM status structure has two perspectives one team centric mimicking our team and department structure and one solution centric focusing on the solutions we bring to the market.

The figure below illustrates the ASDM status structure.

3.1.1 Team status

Team status contains the team self-assessment of its ASDM maturity, metrics related to their security analysis activities as well as an aggregation of the security status of the components they are responsible for.



Axis defines the ASDM maturity as the ASDM version the team currently uses. Since the ASDM is evolving we have defined ASDM versioning where each version of the ASDM contains a unique set of activities. For example, our first version of the ASDM is focused on threat modelling.

Axis has defined the following ASDM versions:

ASDM version	New activities
ASDM 1.0	Risk assessment and threat modelling
ASDM 2.0	Static code review
ASDM 2.1	Privacy by design
ASDM 2.2	Software composition analysis
ASDM 2.3	External penetration testing
ASDM 2.4	Vulnerability scanning and fire drill
ASDM 2.5	Product/Solution security status

Giving the team ownership of which ASDM version they use means that it is the line manager who is responsible for the adoption of new ASDM versions. So instead of a setup where SSG pushes a central ASDM rollout plan it now becomes pull based and controlled by the managers.

3.1.2 Component status

We have a broad definition of component since we need to cover all sorts of architectural entities ranging from Linux demons in the platform, through server software all the way to cloud (micro) services.

Each team must make up their own mind of an abstraction level that works for them in their environment and architecture. As a rule of thumb, teams should avoid inventing a new abstraction level and keep whatever they are already using in their daily work.

The idea is that each team should have a clear view of all their high-risk components, which includes new as well as legacy components. The motivation for this increased interest in legacy components is linked to our ability to look at the security status for solutions. In the case of a solution, we want to have visibility into the security status of all parts of the solution new as well as old.

In practice this means that every team must look at their inventory of components and make a risk assessment.

The first thing we need to know is whether the component has undergone security analysis. If it hasn't, we really don't know anything about the security quality of the component.

We call this property coverage and have defined the following coverage levels:

Coverage	**Description**
Analysis not done	The component has not yet been analyzed
Analysis ongoing	The component is being analyzed
Analysis done	The component has been analyzed

The metrics we use to capture the security quality of the component are based on the security work items in the backlog that are linked to the component. This can be countermeasures that have not been implemented, test cases that have not been executed and security bugs that have not been addressed.

3.1.3 Solution status

Solution status aggregates security status for a set of components that make up the solution.

The first part of the solution status is the analysis coverage of the components. This helps solution owners understand if the security status of the solution is known or if it is not. In one perspective it helps identify the blind spots.

The rest of the solution status contains metrics that capture the security quality of the solution. We do that by looking at the security work items that are linked to the components in the solution.

An important aspect of the security status is the bug bar defined by the solution owners. The solution owners must define an appropriate security level for their solution. For example, this means that the solution should have no outstanding critical or high severity work items open when released to the market.

4. ASDM activities

4.1 Risk assessment

The main purpose with risk assessment is to filter out what development activities that also will require security work within the team.

Risk assessment is done by judging if a new product or added/modified feature in existing products increases the risk exposure. Note that this also includes data privacy aspects and compliance requirements. Examples of changes that have risk impact are new APIs, changes to authorization requirements, new middleware, etc.

4.2 Data privacy

Trust is a key focus area for Axis and, as such, it is important to follow best practices when working with private data collected by our products, solutions and services.

The scope for Axis efforts related to data privacy are defined such that we can:

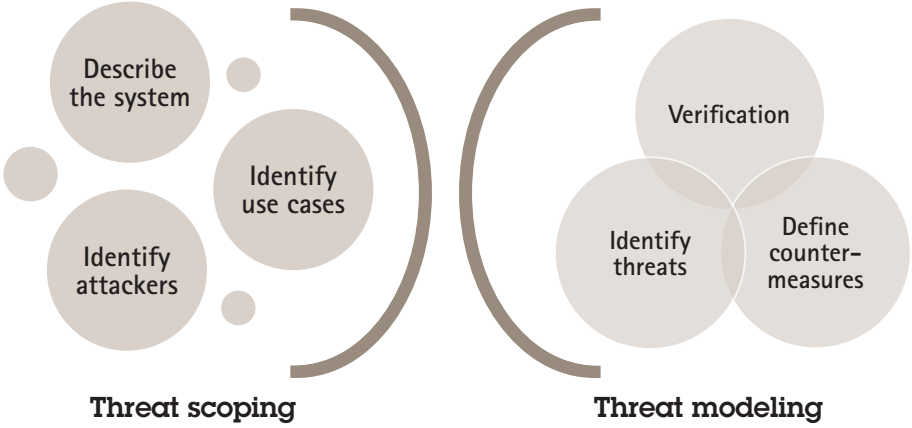
> Fulfill legal obligations
> Fulfill contractual obligations
> Help customers fulfill their obligations

We divide the 'Data privacy' activity into two sub-activities:

> Data privacy assessment
  - Done during 'Risk assessment'
  - Identifies if data privacy analysis is needed
> Data privacy analysis
  - Done, when applicable, during 'threat modeling'
  - Identifies personal data and threats to personal data
  - Defines privacy requirements

4.3 Threat modeling

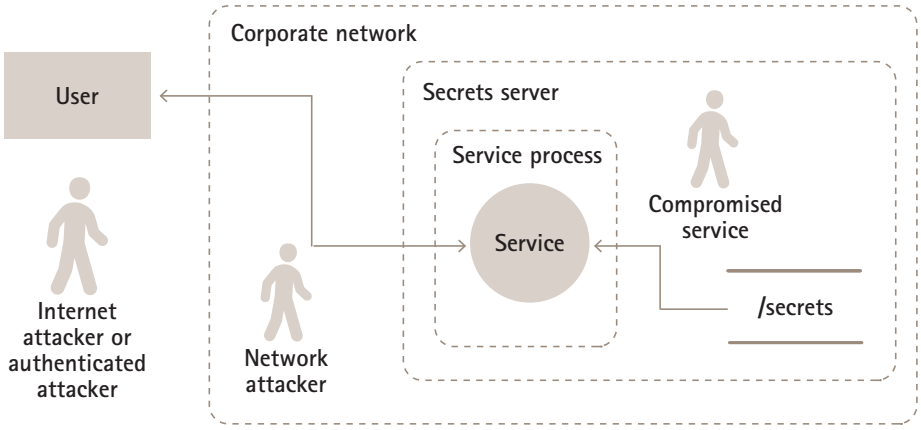
Before we start identifying threats, we need to decide on the scope of the threat model. A way of articulating the scope is to describe the attackers we need to consider. This approach will also allow us to identify the high-level attack surfaces we must include in the analysis.



Focus during threat scoping is on finding and categorizing attackers we want to handle using a high-level description of the system. Preferably the description is done using a data flow diagram (DFD) since it makes it easier to relate the more detailed use case descriptions that are used when doing the threat model.

This does not mean that all the attackers we identify need to be considered, it simply means that we are explicit and consistent on the attackers we will address in the threat model. So, essentially the attackers we choose to consider will define the security level of the system we are assessing.

Note that our attacker description does not factor in attacker capabilities or motivation. We have chosen this approach to simplify and streamline threat modeling as much as possible.



Threat modeling has three steps that can be iterated as the team sees fit:

1. Describe the system using a set of DFDs
2. Use the DFDs to identify threats and describe them in an abuse-case style
3. Define countermeasures and verification for the threats

The result of a threat modeling activity is a threat model that contains prioritized threats and countermeasures. Development work required to address countermeasures is managed by creation of Jira tickets both for the implementation and verification of the countermeasure.

Confluence

ID	Threat	Severity	Countermeasure	Implementation status	Verification	Verification status
UC1:1	A man in the middle reads the password in order to impersonate the user	High	TLS Req.: TLS Best Practices	✓ ~~SSG-376~~ - Enable TLS **DONE**	QART Non-TLS connection should fail Should only be able to connect with recommended cipher suites	✓ **SSG-379** - Test that only recommended cipher suites are possible **NEW** ✓ **SSG-378** - Test non-TLS connection failure **NEW**

Jira Software

Project/XXX-YYY
Priority: High
Tags: Security, SDMActivity, ThreatModel
COMPONENT

Description: Enable TLS

Jira Software

Project/XXX-ZZZ
Priority: Medium
Tags: Security, SDMActivity, ThreatModel
COMPONET

Description: Test TSL

4.4 Static code analysis

In the ASDM teams can use static code analysis in three ways:

> Developer workflow: developers analyze the code they are working on
> Gerrit workflow: developers get feedback in Gerrit
> Legacy workflow: teams analyze high risk legacy components

4.5 Vulnerability scanning

Regular vulnerability scanning allows the development teams to identify and patch software vulnerabilities before products are released to the public, reducing the customer's risk when deploying the product or service. Scanning is performed prior to each release (hardware, software) or on a running schedule (services) using both open-source and commercial vulnerability scanning packages.

The results of the scans are used to generate tickets in the Jira issue tracking platform. Tickets are given a special tag to be identifiable by development teams as coming from a vulnerability scan and that they should be given an elevated priority. All vulnerability scans and Jira tickets are stored centrally for traceability and auditing purposes.

Critical vulnerabilities should be resolved prior to release or in a special service release with other, non-critical vulnerabilities, tracked and resolved in alignment with the firmware or software release cycle. For more information on how vulnerabilities are scored and managed, see the "Vulnerability Management" section.

4.6 External penetration testing

In select cases, third-party penetration testing is performed on Axis hardware or software products. The main purpose of running these tests is to provide insight and assurance regarding the security of the platform at a particular timepoint and for a particular scope.

One of our primary goals with the ASDM is transparency so we encourage our customers to perform external penetration testing on our products and we are happy to collaborate when defining appropriate parameters for testing as well as discussions around interpreting the results.

4.7 Vulnerability management

Axis, since 2021, is a registered CVE naming authority (CNA) and therefore capable of publishing standard CVE reports to the MITRE database for consumption by third-party vulnerability scanners and other tools.

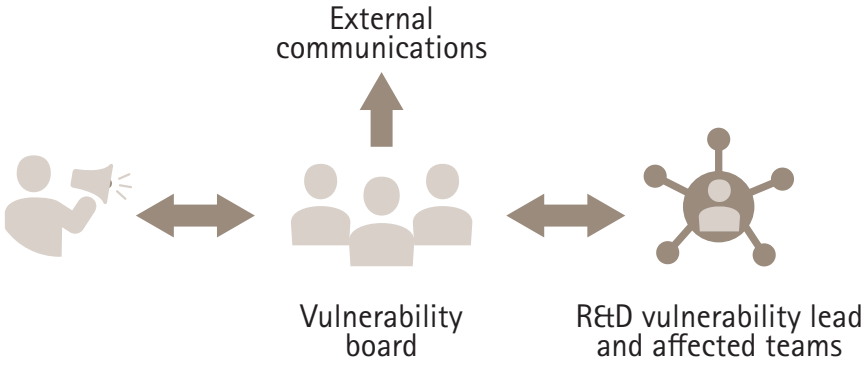
The vulnerability board, VB, is the internal Axis contact point for vulnerabilities discovered by external researchers. Reporting of discovered vulnerabilities and subsequent remediation plans are communicated via the product-security@axis.com email address.

The main VB responsibility is to analyze and prioritize reported vulnerabilities from a business perspective, based on:

> Technical classification provided by the SSG
> Potential risk for end-users in the environment Axis device operates
> Availability of compensating security controls (alternative risk mitigation without patching)

And register the CVE number and work with the reporter to assign a CVSS score to the vulnerability.

VB also drives external communication to partners and customers through the Axis security notification service, press releases, and news articles.



About Axis Communications

Axis enables a smarter and safer world by creating network solutions that provide insights for improving security and new ways of doing business. As the industry leader in network video, Axis offers products and services for video surveillance and analytics, access control, intercom and audio systems. Axis has more than 3,800 dedicated employees in over 50 countries and collaborates with partners worldwide to deliver customer solutions. Axis was founded in 1984 and has its headquarters in Lund, Sweden.

For more information about Axis, please visit our website www.axis.com

©2020-2022 Axis Communications AB. AXIS COMMUNICATIONS, AXIS, ARTPEC and VAPIX are registered trademarks of Axis AB in various jurisdictions. All other trademarks are the property of their respective owners.