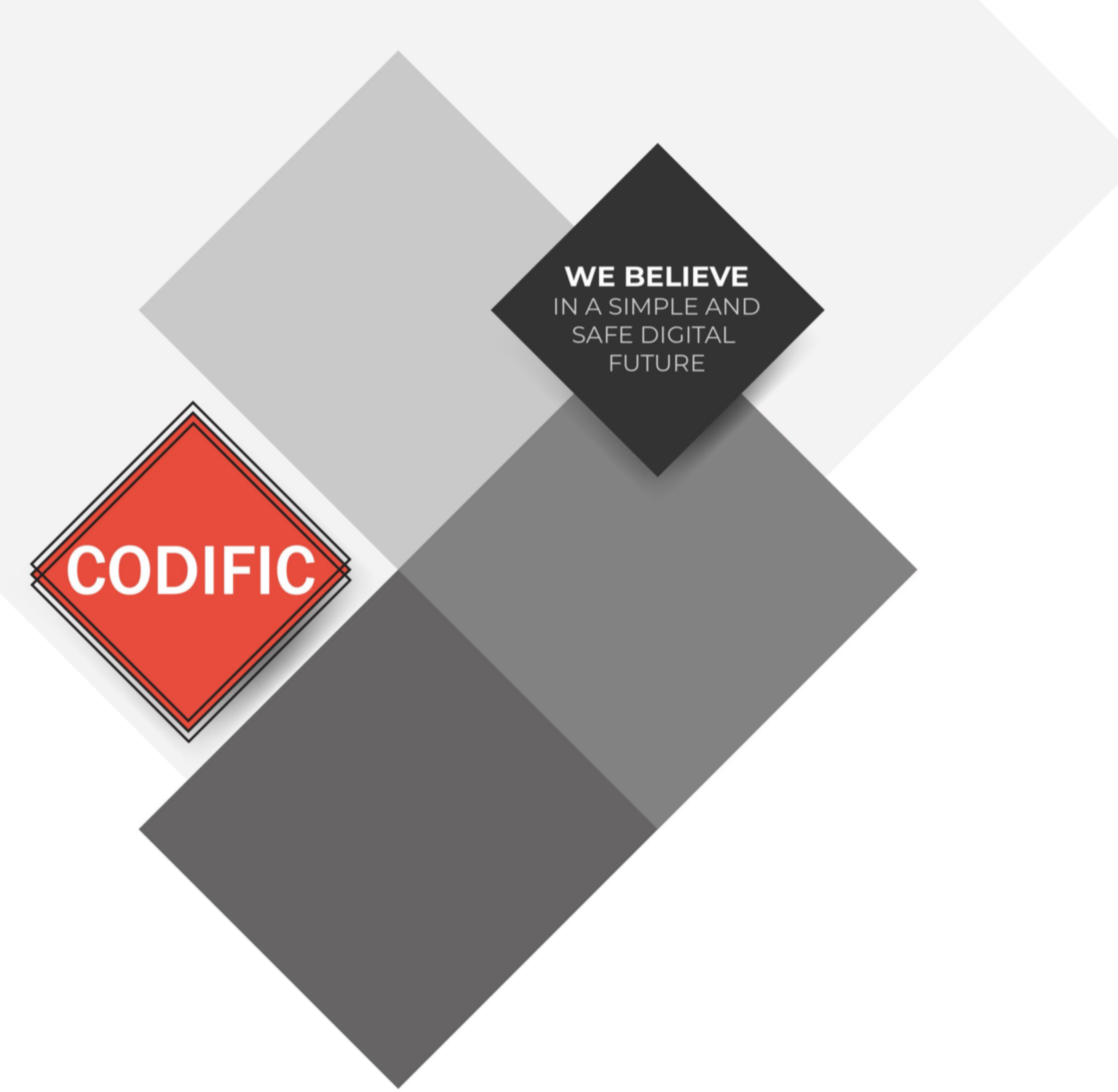


SAMM Assessment Report

Demo Org Playground: Mobile applications

Report date: 20 Feb 2025

Table of Contents

Introduction

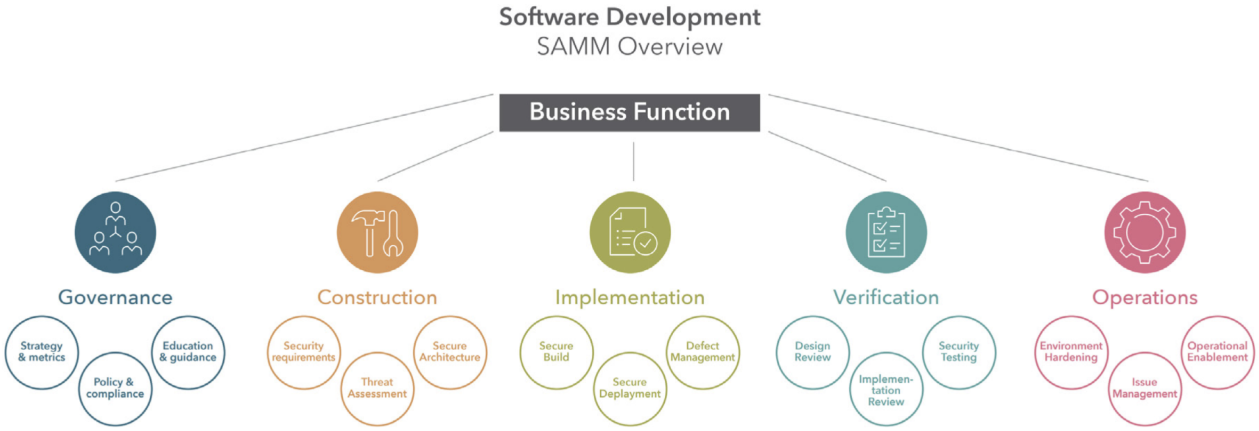
SAMM Assessment Report

This report provides an in-depth analysis of your organization's security posture based on OWASP SAMM. OWASP Software Assurance Maturity Model is a community-led open-source framework that provides an effective and measurable way for organizations and individual business units to analyze and improve their software security posture.

This report was automatically generated using the SAMMY tool. SAMMY is a Software Assurance Maturity Model management tool. It enables organizations to formulate and implement a security assurance program tuned to the risks they are facing. That way other companies can help us build a simple and safe digital future.

The OWASP SAMM Model

SAMM is a prescriptive model, an open framework, which is simple to use and measurable. The solution details are easy enough to follow even for non-security personnel. It helps organizations analyze their current software security practices, build a security program in defined iterations, show progressive improvements in secure practices, and define and measure security-related activities.
SAMM was defined with flexibility in mind so that small, medium, and large organizations using any style of development can customize and adopt it. It provides a means of knowing where your organization is on its journey towards software assurance and understanding what is recommended to move to the next level of maturity.
SAMM does not insist that all organizations achieve the maximum maturity level in every category.



20 Feb 2025

Assessment results for target posture SDLC (High risk applications)

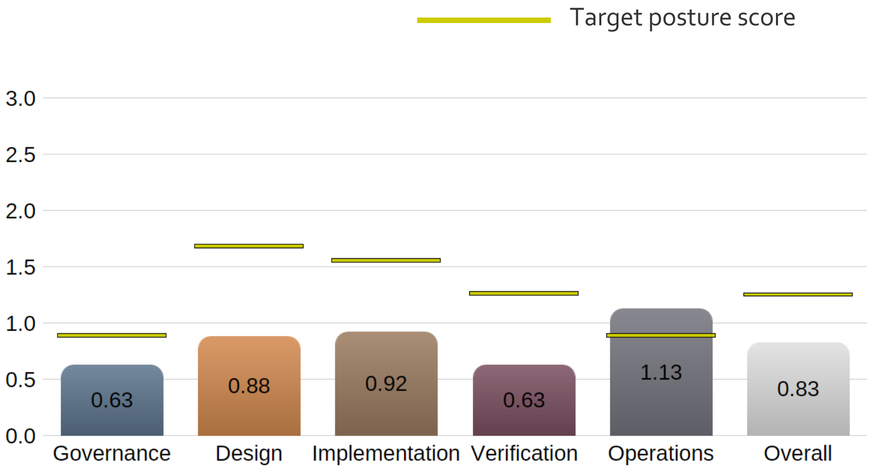
Your overall percentage to target is 60 %.

	Elements	Solution Rating	Target Score	Within target	Percentage to target
	Governance				
	Strategy and Metrics	0.25	0.50		
G-SM-A	Create and Promote	0.25	0.25	Yes	100 %
G-SM-B	Measure and Improve	0.25	0.75	No	33 %
	Policy and Compliance	0.50	1.13		
G-PC-A	Policy and Standards	0.50	1.25	No	40 %
G-PC-B	Compliance Management	0.50	1.00	No	50 %
	Education and Guidance	1.13	1.00		
G-EG-A	Training and Awareness	1.25	1.25	Yes	100 %
G-EG-B	Organization and Culture	1.00	0.75	No	67 %
	Design				
	Threat Assessment	1.00	2.00		
D-TA-A	Application Risk Profile	1.25	1.75	No	71 %
D-TA-B	Threat Modeling	0.75	2.25	No	33 %
	Security Requirements	0.75	1.25		
D-SR-A	Software Requirements	1.25	1.00	Yes	100 %
D-SR-B	Supplier Security	0.25	1.50	No	17 %
	Security Architecture	0.88	1.75		
D-SA-A	Architecture Design	1.00	1.50	No	67 %
D-SA-B	Technology Management	0.75	2.00	No	38 %
	Implementation				
	Secure Build	1.13	1.75		
I-SB-A	Build Process	1.25	2.25	No	56 %
I-SB-B	Software Dependencies	1.00	1.25	No	80 %
	Secure Deployment	1.00	1.88		
I-SD-A	Deployment Process	1.00	1.75	No	56 %
I-SD-B	Secret Management	1.00	2.00	No	50 %
	Defect Management	0.63	1.00		
I-DM-A	Defect Tracking	1.00	1.00	Yes	100 %
I-DM-B	Metrics and Feedback	0.25	1.00	No	25 %
	Verification				
	Architecture Assessment	0.50	0.88		
V-AA-A	Architecture Validation	0.25	0.50	No	50 %
V-AA-B	Architecture Mitigation	0.75	1.25	No	40 %
	Requirements-driven Testing	0.63	0.88		
V-RT-A	Control Verification	1.25	1.50	No	83 %
V-RT-B	Misuse/Abuse Testing	0.00	0.25	No	0 %

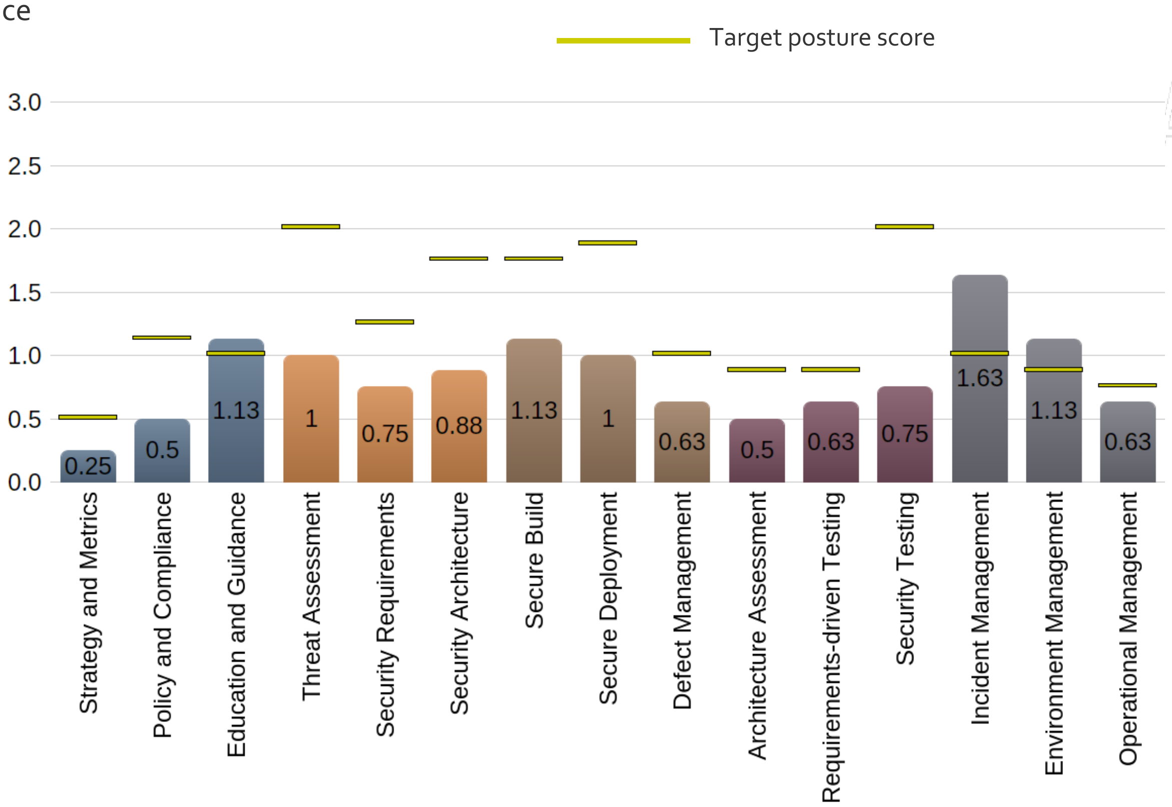
	Security Testing	0.75	2.00		
V-ST-A	Scalable Baseline	1.25	1.75	No	71 %
V-ST-B	Deep Understanding	0.25	2.25	No	11 %
	Operations				
	Incident Management	1.63	1.00		
O-IM-A	Incident Detection	2.00	1.00	Yes	100 %
O-IM-B	Incident Response	1.25	1.00	Yes	100 %
	Environment Management	1.13	0.88		
O-EM-A	Configuration Hardening	1.25	0.75	Yes	100 %
O-EM-B	Patching and Updating	1.00	1.00	Yes	100 %
	Operational Management	0.63	0.75		
O-OM-A	Data Protection	1.00	1.00	Yes	100 %
O-OM-B	System Decommissioning / Legacy Management	0.25	0.50	No	50 %

Current scores

Per business function



Per practice



Governance	0.63

Strategy and Metrics	0.25
G-SM-A Create and Promote	**0.25**
L1: Do you understand the enterprise-wide risk appetite for your applications?	Yes, it covers general risks 0.25 (target score: 0.25)
L2: Do you have a strategic plan for application security and use it to make decisions?	No 0.00 (target score: 0.00)
L3: Do you regularly review and update the Strategic Plan for Application Security?	No 0.00 (target score: 0.00)
Observations • There is a very basic enterprise-wide strategic plan in place taking into account the very basic risks.	
Target posture reference documentation	
It is essential to have some basic understanding of the enterprise risk appetite even for SMEs. However anything beyond the basics is typically far fetched for smaller companies.	
G-SM-B Measure and Improve	**0.25**
L1: Do you use a set of metrics to measure the effectiveness and efficiency of the application security program across applications?	Yes, for one metrics category 0.25 (target score: 0.50)
L2: Did you define Key Performance Indicators (KPI) from available application security metrics?	No 0.00 (target score: 0.25)
L3: Do you update the Application Security strategy and roadmap based on application security metrics and KPIs?	No 0.00 (target score: 0.00)
Observations • It seems only level of effort metrics are tracked across the organization, however we haven't observed any KPIs in place.	
Target posture reference documentation	
It makes sense to have at least some metrics in place. We recommend having at least number of security defects in place and perhaps the level of effort metrics. Based on these basic KPIs could be set.	

Policy and Compliance	0.50
G-PC-A Policy and Standards	**0.5**
L1: Do you have and apply a common set of policies and standards throughout your organization?	Yes, for at least half of the applications 0.50 (target score: 1.00)
L2: Do you publish the organization's policies as test scripts or run-books for easy interpretation by development teams?	No 0.00 (target score: 0.25)
L3: Do you regularly report on policy and standard compliance, and use that information to guide compliance improvement efforts?	No 0.00 (target score: 0.00)
Observations • There are basic common policies and standards in place available throughout the organization. However only half of the applications comply with the policy.	
Target posture reference documentation	
For smaller organizations the common set of policies as well as standards are likely to be limited, but it is essential to have them in place. It also makes sense to have some of them translated into verification checklists and test scripts.	
G-PC-B Compliance Management	**0.5**
L1: Do you have a complete picture of your external compliance obligations?	Yes, for at least half of the applications 0.50 (target score: 1.00)
L2: Do you have a standard set of security requirements and verification procedures addressing the organization's external compliance obligations?	No 0.00 (target score: 0.00)
L3: Do you regularly report on adherence to external compliance obligations and use that information to guide efforts to close compliance gaps?	No 0.00 (target score: 0.00)
Observations • The only external compliance obligation this team faces is GDPR.	
Target posture reference documentation	
In today's application landscape any organization large or small should comply with some regulations (e.g., GDPR). Having a clear view of the compliance obligations is a must.	

Education and Guidance	1.13
G-EG-A Training and Awareness	**1.25**
L1: Do you require employees involved with application development to take SDLC training?	Yes, most or all of them 1.00 (target score: 1.00)
L2: Is training customized for individual roles such as developers, testers, or security champions?	Yes, for some of the training 0.25 (target score: 0.25)
L3: Have you implemented a Learning Management System or equivalent to track employee training and certification processes?	No 0.00 (target score: 0.00)
Observations • There is a solid set of mandatory training for this team in place. • Only some roles have a customized training, namly the security champions. No training completion tracking system is implemented.	
Target posture reference documentation	
There should be a mandatory security training for everyone involved in the SDLC. Beyond that the security champions should have more customized training.	
G-EG-B Organization and Culture	**1**
L1: Have you identified a Security Champion for each development team?	Yes, for most or all of the teams 1.00 (target score: 0.50)
L2: Does the organization have a Secure Software Center of Excellence (SSCE)?	No 0.00 (target score: 0.00)
L3: Is there a centralized portal where developers and application security professionals from different teams and business units are able to communicate and share information?	No 0.00 (target score: 0.25)
Observations • Every team has a dedicated security champion that is a developer with more security expertise.	
Target posture reference documentation	
Having security savvy team members across all development teams is the first step towards bringing security awareness and culture to your organization (large or small). This is relatively easy to attain for smaller organizations. Once there is a security effort in place it is also a good idea to start centralizing the knowledge into a centralized portal. The portal can be a simple Wiki. However do make sure the wiki is pragmatic and actually being used by the team members.	

Design	0.88

Threat Assessment	1.00
D-TA-A Application Risk Profile	**1.25**
L1: Do you classify applications according to business risk based on a simple and predefined set of questions?	Yes, most or all of them 1.00 (target score: 1.00)
L2: Do you use centralized and quantified application risk profiles to evaluate business risk?	Yes, for some applications 0.25 (target score: 0.50)
L3: Do you regularly review and update the risk profiles for your applications?	No 0.00 (target score: 0.25)

Target posture reference documentation

In essence this practice requires you to create a systematic checklist that quantifies all your applications for business risk. This can (and should) be something very pragmatic so that everyone within the organization can discern which applications are more and less important in terms of risk. If you have a single application it still make sense to do this exercise because everyone in the organization needs to understand the impact a negative event could have on the business. Note that simply stating that a breach is a bad thing to happen is not enough. You need to think about the type of breaches and their impact.

Make sure to review the risk profiles from time to time.

D-TA-B Threat Modeling	0.75
L1: Do you identify and manage architectural design flaws with threat modeling?	Yes, at least half of them 0.50 (target score: 1.00)
L2: Do you use a standard methodology, aligned on your application risk levels?	Yes, for some applications 0.25 (target score: 1.00)
L3: Do you regularly review and update the threat modeling methodology for your applications?	No 0.00 (target score: 0.25)

Target posture reference documentation

This activity is actually part of your threat modeling efforts. Here is a description of a pragmatic threat modeling process that ties threat modeling, architecture validation and mitigation practices.

Plan regular threat modeling sessions done in a group.

- Define a note-taker for the session who will keep a bullet list of notes.
- If this is not your first session it is up to you to decide whether you want to start from where you left off. There are advantages and disadvantages for doing that.
- Create a high-level overview of the application's architecture by going over the main security controls and discussing how these tackle certain security threats.
- From there onwards move to a Data Flow Diagram and consider how the data flows across your system. The team should switch to an "attacker mindset" from this point on and start coming up with potential threats. Any threat mentioned should be noted and at this stage you should not yet be focusing on the solution.
- Once the list of threats is in place move on towards looking for how your application (and its architecture) currently handles the issue.
- Define new controls / solutions if necessary to tackle some of the threats.
- Once the controls are defined go over the modified architecture and solution and discuss once again with the team whether this will tackle the threats efficiently.
- A curated list of the threats should always end up in your issue tracking system especially for items that you will accept in terms of risk.

Note that your threat modeling sessions do not have to cover the complete application. Rather you could decide to focus on a specific part of the application (e.g., the mobile app API, the access control logic, etc).

Security Requirements	0.75
D-SR-A Software Requirements	**1.25**
L1: Do project teams specify security requirements during development?	Yes, for most or all of the applications 1.00 (target score: 1.00)
L2: Do you define, structure, and include prioritization in the artifacts of the security requirements gathering process?	Yes, some of the time 0.25 (target score: 0.00)
L3: Do you use a standard requirements framework to streamline the elicitation of security requirements?	No 0.00 (target score: 0.00)
Target posture reference documentation	
Security requirements should be part of the regular development process. OWASP ASVS is the perfect place to start. You should be pragmatic about it and start from a relatively limited set of requirements from ASVS. The more you mature the more requirements you should start pulling in.	
D-SR-B Supplier Security	**0.25**
L1: Do stakeholders review vendor collaborations for security requirements and methodology?	Yes, some of the time 0.25 (target score: 1.00)
L2: Do vendors meet the security responsibilities and quality measures of service level agreements defined by the organization?	No 0.00 (target score: 0.50)
L3: Are vendors aligned with standard security controls and software development tools and processes that the organization utilizes?	No 0.00 (target score: 0.00)
Target posture reference documentation	
If you are working with outsourced development teams it is essential to make security part of the contractual agreements with those suppliers. Otherwise you should be scoring "No" for all answers.	

Security Architecture	0.88
D-SA-A Architecture Design	**1**
L1: Do teams use security principles during design?	Yes, for at least half of the applications 0.50 (target score: 1.00)
L2: Do you use shared security services during design?	Yes, for some applications 0.25 (target score: 0.25)
L3: Do you base your design on available reference architectures?	Yes, for some applications 0.25 (target score: 0.25)
Target posture reference documentation	
The targets in this stream largely depend on the nature of your organization's business. If you are developing just one application your target posture should be Yes, No, No for the three levels respectively. The same set of target scores is also relevant if you have several applications, but none of them share the underlying technology. However as soon as you have more than one application in place that leverage the same underlying technology you should absolutely make sure you have a reference architecture and shared security services in place.	
D-SA-B Technology Management	**0.75**
L1: Do you evaluate the security quality of important technologies used for development?	Yes, for at least half of the applications 0.50 (target score: 1.00)
L2: Do you have a list of recommended technologies for the organization?	Yes, for some of the technology domains 0.25 (target score: 1.00)
L3: Do you enforce the use of recommended technologies within the organization?	No 0.00 (target score: 0.00)
Target posture reference documentation	
You must evaluate the security quality of all technologies that you leverage. You should also have a set of technologies everyone should use. Note that this stream does not deal with any third party dependencies. However we are not only talking about your core technologies (e.g. .NET, Java, Javascript, etc), but also the basic frameworks and components of those frameworks. For instance, if you are developing in Angular your team might decide to use Axios although all projects are written in the built-in HttpClient (I know it might feel like a third party dependency story, but it's not). So you need to have a list of technologies and stick to that list. Anything being added should pass a rigorous review process focusing on amongst others security.	

Implementation	0.92

Secure Build	1.13
I-SB-A Build Process	**1.25**
L1: Is your full build process formally described?	Yes, for most or all of the applications 1.00 (target score: 1.00)
L2: Is the build process fully automated?	Yes, for some applications 0.25 (target score: 1.00)
L3: Do you enforce automated security checks in your build processes?	No 0.00 (target score: 0.25)
Target posture reference documentation	
It is hard to think of a scenario where you do not want to have fully automated builds. For mobile apps this might be somewhat challenging, but it is still possible. You want to also consider adding automated security checks to your build processes. Note that randomly adding SAST and SCA tooling is perhaps not the best idea. So you should be looking for a nice trade-off between adding some automated security and making sure your new process doesn't introduce too many gates (especially knowing that tools might block things with false positives).	
I-SB-B Software Dependencies	**1**
L1: Do you have solid knowledge about dependencies you're relying on?	Yes, for most or all of the applications 1.00 (target score: 1.00)
L2: Do you handle 3rd party dependency risk by a formal process?	No 0.00 (target score: 0.25)
L3: Do you prevent build of software if it's affected by vulnerabilities in dependencies?	No 0.00 (target score: 0.00)
Target posture reference documentation	
Knowing what dependencies you are leveraging is a must-have. However that is not enough. You should have some process in place to vet the new dependencies (as well as those that you already have in place). Note that the formal review from a legal perspective is also a good idea (i.e., copyleft licenses that might have a profound impact on your business). However it is critical to have at least a very basic checklist that discerns between an acceptable and unacceptable dependency (aside from the fact whether it contains known vulnerabilities or not).	

Secure Deployment	1.00
I-SD-A Deployment Process	**1**
L1: Do you use repeatable deployment processes?	Yes, for most or all of the applications 1.00 (target score: 1.00)
L2: Are deployment processes automated and employing security checks?	No 0.00 (target score: 0.50)
L3: Do you consistently validate the integrity of deployed artifacts?	No 0.00 (target score: 0.25)

Target posture reference documentation

You should try to automate the deploy processes as much as possible to reduce the chance of any manual errors. For a typical web application deployment involves replacing the running container with the new container image. This is typically done by a script (which could be a CI/CD pipeline). Hence, from a technological perspective this is relatively straightforward and there are many options for scripting. However it is essential to harden the complet deployment pipeline. Here is a list of items that are must have:

- Only specific people from the team should have deploy rights.
- Deploy to production should not be automated (staging is fine).
- You should have a log in place for past deployments.
- You should sign your images at build time and verify the signature at deploy time. Cosign is a great place to look for the specifics of that.

For mobile applications things might seem a bit more complicated, but there are platforms and tools to help you out with that. Fastlane open source project is perhaps where you would like to start.

I-SD-B Secret Management	**1**
L1: Do you limit access to application secrets according to the least privilege principle?	Yes, for most or all of the applications 1.00 (target score: 1.00)
L2: Do you inject production secrets into configuration files during deployment?	No 0.00 (target score: 1.00)
L3: Do you practice proper lifecycle management for application secrets?	No 0.00 (target score: 0.00)

Target posture reference documentation

There are actually various places where secret management is essential across the SDLC.

CI/CD

Secret management in the build and deploy pipelines should follow best practices set by your automation tooling. E.g., for GitLab you should use the CI/CD variables for that.

Secrets needed by the application

For any secrets your application needs (e.g., database access, third party API access) you should leverage solutions like HashiCorp Vault, GCP Key Manager, etc. that enable you to inject these secrets at deploy time in the application container configuration. Ideally you also have a secret scanning solution that looks for hard-coded secrets in the source code.

Secrets to access infrastructure

The secrets to access the infrastructure are typically with the operations team. You need to make sure that the number of people who can access the infrastructure is limited (principles of least privilege and need-to-know). Furthermore, people who do have access need to use personal password managers and store their keys there. Stronger MFA solutions are a must have (e.g., Yubikey, SSH keys protected by a password, etc).

Defect Management	0.63
I-DM-A Defect Tracking	**1**
L1: Do you track all known security defects in accessible locations?	Yes, for most or all of the applications 1.00 (target score: 1.00)
L2: Do you keep an overview of the state of security defects across the organization?	No 0.00 (target score: 0.00)
L3: Do you enforce SLAs for fixing security defects?	No 0.00 (target score: 0.00)
Target posture reference documentation	
Ideally, you should bring all the defects into a centralized place. JIRA is typically a standard platform for large organizations. However in our experience, YouTrack is probably the best platform out there in terms of UI as well as performance (developed by JetBrains). Make sure to annotate all security related issues with a label. It is a good idea to be able to pull a report of e.g., all defects found during pen testing or all security related bugs. This will help you follow up on how well you are doing in terms of security.	
I-DM-B Metrics and Feedback	**0.25**
L1: Do you use basic metrics about recorded security defects to carry out quick win improvement activities?	Yes, for some applications 0.25 (target score: 1.00)
L2: Do you improve your security assurance program upon standardized metrics?	No 0.00 (target score: 0.00)
L3: Do you regularly evaluate the effectiveness of your security metrics so that its input helps drive your security strategy?	No 0.00 (target score: 0.00)
Target posture reference documentation	
Ideally, your SAMM implementation efforts as well as prioritization should be metrics-driven. Unfortunately, that is not as easy as it sounds in reality. The most practical tips when it comes to metrics would be to set specific goals and start from there rather than look for metrics and imagine what they could be used for. Goal-Question-Metric framework is a great theoretical framework to leverage for this. Here is a list of some useful goals especially for smaller companies: • Goal: minimize the number of security regressions (i.e., security issues found previously that reoccur). • Goal: maximize the learning from security issues. • Goal: minimize the number of exploitable vulnerabilities in production.	

Verification	0.63

Architecture Assessment	0.50
V-AA-A Architecture Validation	**0.25**
L1: Do you review the application architecture for key security objectives on an ad-hoc basis?	Yes, for some applications 0.25 (target score: 0.50)
L2: Do you regularly review the security mechanisms of your architecture?	No 0.00 (target score: 0.00)
L3: Do you regularly review the effectiveness of the security controls?	No 0.00 (target score: 0.00)

Target posture reference documentation

This activity is actually part of your threat modeling efforts. Here is a description of a pragmatic threat modeling process that ties threat modeling, architecture validation and mitigation practices.

Plan regular threat modeling sessions done in a group.

- Define a note-taker for the session who will keep a bullet list of notes.
- If this is not your first session it is up to you to decide whether you want to start from where you left off. There are advantages and disadvantages for doing that.
- Create a high-level overview of the application's architecture by going over the main security controls and discussing how these tackle certain security threats.
- From there onwards move to a Data Flow Diagram and consider how the data flows across your system. The team should switch to an "attacker mindset" from this point on and start coming up with potential threats. Any threat mentioned should be noted and at this stage you should not yet be focusing on the solution.
- Once the list of threats is in place move on towards looking for how your application (and its architecture) currently handles the issue.
- Define new controls / solutions if necessary to tackle some of the threats.
- Once the controls are defined go over the modified architecture and solution and discuss once again with the team whether this will tackle the threats efficiently.
- A curated list of the threats should always end up in your issue tracking system especially for items that you will accept in terms of risk.

Note that your threat modeling sessions do not have to cover the complete application. Rather you could decide to focus on a specific part of the application (e.g., the mobile app API, the access control logic, etc).

V-AA-B Architecture Mitigation	0.75
L1: Do you review the application architecture for mitigations of typical threats on an ad-hoc basis?	Yes, for some applications 0.25 (target score: 1.00)
L2: Do you regularly evaluate the threats to your architecture?	Yes, for some applications 0.25 (target score: 0.25)
L3: Do you regularly update your reference architectures based on architecture assessment findings?	Yes, for some applications 0.25 (target score: 0.00)

Target posture reference documentation

This activity is actually part of your threat modeling efforts. Here is a description of a pragmatic threat modeling process that ties threat modeling, architecture validation and mitigation practices.

Plan regular threat modeling sessions done in a group.

- Define a note-taker for the session who will keep a bullet list of notes.
- If this is not your first session it is up to you to decide whether you want to start from where you left off. There are advantages and disadvantages for doing that.
- Create a high-level overview of the application's architecture by going over the main security controls and discussing how these tackle certain security threats.
- From there onwards move to a Data Flow Diagram and consider how the data flows across your system. The team should switch to an "attacker mindset" from this point on and start coming up with potential threats. Any threat mentioned should be noted and at this stage you should not yet be focusing on the solution.
- Once the list of threats is in place move on towards looking for how your application (and its architecture) currently handles the issue.
- Define new controls / solutions if necessary to tackle some of the threats.
- Once the controls are defined go over the modified architecture and solution and discuss once again with the team whether this will tackle the threats efficiently.
- A curated list of the threats should always end up in your issue tracking system especially for items that you will accept in terms of risk.

Note that your threat modeling sessions do not have to cover the complete application. Rather you could decide to focus on a specific part of the application (e.g., the mobile app API, the access control logic, etc).

Requirements-driven Testing	0.63
V-RT-A Control Verification	**1.25**
L1: Do you test applications for the correct functioning of standard security controls?	Yes, most or all of them 1.00 (target score: 1.00)
L2: Do you consistently write and execute test scripts to verify the functionality of security requirements?	Yes, some of them 0.25 (target score: 0.25)
L3: Do you automatically test applications for security regressions?	No 0.00 (target score: 0.25)
Target posture reference documentation	
This stream ties into the security requirements stream. Once you have started explicitly describing the security requirements, they should become part of your sprint planning. Anything on your sprint planning should be tested. Ideally both by unit/integration tests as well as manual QA. Hence, requirements testing is about going over the security requirements and making sure they are correctly implemented.	
V-RT-B Misuse/Abuse Testing	**0**
L1: Do you test applications using randomization or fuzzing techniques?	No 0.00 (target score: 0.25)
L2: Do you create abuse cases from functional requirements and use them to drive security tests?	No 0.00 (target score: 0.00)
L3: Do you perform denial of service and security stress testing?	No 0.00 (target score: 0.00)
Target posture reference documentation	
Think of misuse/abuse testing as negative test cases for security requirements. However these typically go beyond a simple negative test cases. • The first activity focuses on automation-driven fuzzing where appropriate tooling (e.g., ZAP) can try to break your application by providing garbled data for all the input fields. If the application crashes then the test was successful. • The second activity is about crafting explicit abuse test cases focusing on a full use case rather than on a specific input field. E.g., hijacking a session ID from a user and changing the credentials so that the attacker effectively hijacks the account. This activity largely ties into threat modeling. • The last maturity level is about stress testing, but once again your efforts should go beyond simple scalability tests for number of users. You should look at specific application areas that could exhaust resources (e.g., rate limiting CPU hungry report generation). Note that the target posture for smaller companies currently only requires implementing the first activity.	

<table>
<tr><th>Security Testing</th><th>0.75</th></tr>
<tr><td><b>V-ST-A Scalable Baseline</b></td><td><b>1.25</b></td></tr>
<tr><td>L1: Do you scan applications with automated security testing tools?</td><td>Yes, most or all of them
1.00 (target score: 1.00)</td></tr>
<tr><td>L2: Do you customize the automated security tools to your applications and technology stacks?</td><td>Yes, some of them
0.25 (target score: 0.50)</td></tr>
<tr><td>L3: Do you integrate automated security testing into the build and deploy process?</td><td>No
0.00 (target score: 0.25)</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2">Target posture reference documentation</td></tr>
<tr><td colspan="2">There is no shortage of security scanners. However in a default setting all these tools are typically going to generate a ton of findings most of which are not really relevant. Here is a list of must-have for this stream.
• For Static Application Security Testing (SAST) tools invest in modifying and adding the rules to match your technology stack and application domain.
• Integrate the tool in your build pipeline and break the build if high priority security issues are found.
• For Dynamic Application Security Testing (DAST) tools invest in fine-tuning them. Run these tools as scheduled jobs (e.g., during nights and weekends). Issue notifications if the tools find anything beyond the security baseline you have specified.</td></tr>
<tr><td><b>V-ST-B Deep Understanding</b></td><td><b>0.25</b></td></tr>
<tr><td>L1: Do you manually review the security quality of selected high-risk components?</td><td>Yes, for some components
0.25 (target score: 1.00)</td></tr>
<tr><td>L2: Do you perform penetration testing for your applications at regular intervals?</td><td>No
0.00 (target score: 0.25)</td></tr>
<tr><td>L3: Do you use the results of security testing to improve the development lifecycle?</td><td>No
0.00 (target score: 1.00)</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2">Target posture reference documentation</td></tr>
<tr><td colspan="2">It is a good idea to hire third parties to run a penetration testing for your applications. Try to scope the pen tests so that you describe the most important assets and high risk components in your applications. It is critical though to have a systematic learning process after a pen test rather than just fixing the issues. Here is a list of items to consider:
• Look for additional security training based on the findings.
• Revisit all other applications and components where similar findings are likely to happen.
• Revisit security requirements and their verification procedures to minimize the risk of (similar) findings reoccurring.</td></tr>
</table>

Operations	1.13

Incident Management	1.63
O-IM-A Incident Detection	**2**
L1: Do you analyze log data for security incidents periodically?	Yes, for most or all of the applications 1.00 (target score: 1.00)
L2: Do you follow a documented process for incident detection?	Yes, for most or all of the applications 1.00 (target score: 0.00)
L3: Do you review and update the incident detection process regularly?	No 0.00 (target score: 0.00)

Target posture reference documentation

Application error logs need to be closely monitored as both regular and security bugs may be lurking there. Here are a few ideas for best practices:

- Periodically revisit your application error logs to look for potential issues. Using a scheduled job create issues and assign to specific responsible team members.
- Consider sending all application error logs via mail to the responsible users.
- Consider fine-tuning the application to skip logging certain events that are clearly false positives.
- If the rate of error logs exceeds a certain threshold for a unit of time (e.g., more than 10 errors per hour) consider issuing notifications to the responsible team or team members. Pushing a notification in Slack or Teams is a good idea.

O-IM-B Incident Response	**1.25**
L1: Do you respond to detected incidents?	Yes, for most or all of the incidents 1.00 (target score: 1.00)
L2: Do you use a repeatable process for incident handling?	Yes, for some incident types 0.25 (target score: 0.00)
L3: Do you have a dedicated incident response team available?	No 0.00 (target score: 0.00)

Target posture reference documentation

For any suspicious activity that could indicate an incident you should have a minimal set of guidelines to investigate it. Make sure to do this within your issue tracking systems to that a documented log persists.

Environment Management	1.13
O-EM-A Configuration Hardening	**1.25**
L1: Do you harden configurations for key components of your technology stacks?	Yes, for most or all of the components 1.00 (target score: 0.50)
L2: Do you have hardening baselines for your components?	Yes, for some components 0.25 (target score: 0.25)
L3: Do you monitor and enforce conformity with hardening baselines?	No 0.00 (target score: 0.00)
Target posture reference documentation	
There are three topics to consider for this stream. 1. Make sure your technology framework and components are hardened. Any technology and framework has best practice hardening guidelines regarding security. The OWASP Cheatsheet Series is a great place to look for them. 2. Ensure your OS components are hardened. If you are leveraging SSH access to the server this is perhaps where you should start. 3. If you are leveraging docker ensure you use best practices for Docker configuration (e.g., not running the docker containers as root).	
O-EM-B Patching and Updating	**1**
L1: Do you identify and patch vulnerable components?	Yes, for most or all of the components 1.00 (target score: 1.00)
L2: Do you follow an established process for updating components of your technology stacks?	No 0.00 (target score: 0.00)
L3: Do you regularly evaluate components and review patch level status?	No 0.00 (target score: 0.00)
Target posture reference documentation	
This SAMM stream typically deals with container images and OS components. You should introduce some regularity in patching them. For instance, a monthly docker image update as well as apt updates for the OS is sufficient to make sure you cover most of the risks. Monitor threat intelligence sources. High risk vulnerabilities are typically making headline news so critical issues are typically hard to miss.	

Operational Management	0.63
O-OM-A Data Protection	**1**
L1: Do you protect and handle information according to protection requirements for data stored and processed on each application?	Yes, for most or all of the applications 1.00 (target score: 1.00)
L2: Do you maintain a data catalog, including types, sensitivity levels, and processing and storage locations?	No 0.00 (target score: 0.00)
L3: Do you regularly review and update the data catalog and your data protection policies and procedures?	No 0.00 (target score: 0.00)
Target posture reference documentation	
Having a clear view on the data and sensitivity levels your applications are processing is actually mandatory in most parts of the world. For level 1 it is sufficient to have an ad-hoc document in place describing the data. However the most essential part of this stream is to ensure as much as possible that your team does NOT have access to production data. There are virtually no scenarios where this is an acceptable risk. As an alternative consider exporting an anonimized version of the database and giving access to this database to specific members of your team (need-to-know principle remains relevant). Ensure deanonimization remains impossible.	
O-OM-B System Decommissioning / Legacy Management	**0.25**
L1: Do you identify and remove systems, applications, application dependencies, or services that are no longer used, have reached end of life, or are no longer actively developed or supported?	Yes, for some applications 0.25 (target score: 0.50)
L2: Do you follow an established process for removing all associated resources, as part of decommissioning of unused systems, applications, application dependencies, or services?	No 0.00 (target score: 0.00)
L3: Do you regularly evaluate the lifecycle state and support status of every software asset and underlying infrastructure component, and estimate their end of life?	No 0.00 (target score: 0.00)