MASTERING SOFTWARE ASSURANCE: EFFECTIVE SAMM ASSESSMENT STRATEGIES

Codific & Toreon
White Paper

OWASP SAMM

Introduction

Ensuring robust security practices is critical in today's software development landscape. The Software Assurance Maturity Model (SAMM) is a comprehensive framework designed to help organizations analyze, improve, and manage their software security posture. This whitepaper provides a practical guide on performing OWASP SAMM assessments effectively.

SAMM's mission is to raise awareness and educate organizations on designing, developing, and deploying secure software. Through its self-assessment model, SAMM enables organizations to measure their current security capabilities, identify areas for improvement, and create a roadmap for future enhancements. SAMM is both technology and process agnostic. It supports the entire software lifecycle and it is adaptable to various organizational needs.

Recognizing that security is not a one-size-fits-all endeavor, SAMM is designed to be evolutive and risk-driven. This whitepaper consolidates best practices, offering a structured approach to conducting SAMM assessments. Organizations can use these tools to benchmark current practices and strategically plan security improvements.

The first step toward improving your organization's software security is to understand your current situation. An initial SAMM assessment provides a comprehensive overview of your security posture. This evaluation highlights strengths and weaknesses and guides the development of a targeted improvement strategy.

This whitepaper explores various assessment tools, including the SAMM Toolbox (available in Microsoft Excel and Google Spreadsheet formats) and online tools like SAMMWise and SAMMY. These resources facilitate data collection and analysis through questionnaires, interviews, and artifact examination. The accuracy and effectiveness of these assessments depend on the assessor's expertise and understanding of application security best practices.

We discuss the two types of SAMM assessments: self-assessment and expert assessment. Self-assessments are cost-effective and allow teams to quickly familiarize themselves with SAMM. Expert assessments, involving internal or external practitioners, provide deeper insights and more precise evaluations.

Additionally, we cover the methodologies of questionnaire-based and interview-based assessments, outlining their benefits and potential challenges. Detailed planning and execution of assessment interviews ensure a comprehensive understanding of the organization's security practices and foster a collaborative atmosphere for continuous improvement.

In summary, this whitepaper serves as a practical guide for organizations aiming to measure and strengthen their software security using the SAMM framework. By following the outlined best practices and leveraging the provided tools and methodologies, organizations can enhance their software development process to solve the application security challenges and become resilient.

SAMM assessment

The primary goal for a SAMM assessment is to evaluate the current software posture. To achieve this goal, you need to gather and evaluate information regarding the various activities described by SAMM. The outcome of a SAMM assessment ideally presents both where the organization is in terms of their security posture and where they should be in terms of their improvement roadmap.

Assessment tools

There are several options when it comes to creating the assessment. We have 2 versions of the SAMM Toolbox, a Microsoft Excel Toolbox and a Google Spreadsheet Toolbox. You can also leverage the SAMMWise or the SAMMY online tools for running SAMM Assessments. We recommend SAMMY. You can then start gathering the information necessary to fill out a SAMM assessment. You can do that through questionnaires, interviews, or by requesting and investigating artifacts (such as code, configuration, documentation, logs). You then need to evaluate this information to answer the SAMM questions accurately. Typically, the more expertise you have in application security, the more accurate your assessment will be. The quality criteria for evaluating SAMM security activities require proper understanding of best practices. Moreover, best practices in web application security might be quite different from those for example, IoT devices.

Assessment types

There are typically two types of SAMM assessments you can perform. Each has its upsides and drawbacks. Ideally, you want to have a good mix of self-assessment and independent expert assessment. You may want to start with an independent expert assessment. As you gain more insights you could move to self-assessments after that.

Self-assessment

There are typically two types of SAMM assessments you can perform. Each has its upsides and drawbacks. Ideally, you want to have a good mix of self-assessment and independent expert assessment. You may want to start with an independent expert assessment. As you gain more insights you could move to self-assessments after that.

Expert assessment

You can also run an expert assessment by inviting either an independent internal assessor or an external SAMM practitioner. The expert assessment will typically take more time and effort, but will end up being more accurate, more thorough and provide better insights when planning the improvement roadmap. The expert assessment will often require an interview format.

Questionnaire-based assessment

A questionnaire-based assessment requires one or multiple stakeholders to go through the list of 90 questions. This can be perceived as overwhelming. There is also some room for interpretation in the assessment questions. Hence answers across the different teams in your organizations might be inconsistent. The goal of SAMM is to know where you are in order to set up the right improvement roadmap. However if you cannot measure precisely, your improvement strategy may not be optimal, especially for improvements that involve multiple teams. Moreover, large questionnaires may lead to questionnaire-fatigue. Teams might be refusing or reluctant to provide answers or give their time and the necessary attention required. After an initial SAMM assessment for a given scope, the questionnaire-based approach is suitable for updating the SAMM scores. The stakeholder responsible for this should have gained sufficient understanding of the model specifics to be able to do that. Furthermore, the number of practices that need a score update is likely to be limited. The reason for that is that SAMM advocates an iterative approach to improvements. Hence, the improvement roadmap should involve only a relatively limited number of activities.

Interview-based assessment

Interviews are a great alternative for questionnaires. Having a conversation can be more appealing than a list of questions. The interviewer can explain the questions, explain the criteria and ask follow-up questions to gain better understanding or to double check answers. Also, in an interview you can invite people to 'open up' or pick up on certain non-verbal signs regarding a specific topic. These things help to identify issues that are handy for the assessment, but also very useful in case the assessment is followed by recommendations to improve. We would recommend adding the interviewer's observations next to each SAMM stream. These notes serve as a documentation of existing security practices within the assessment scope. They can also be helpful when validating the interview with the stakeholders (see the next section). Here are some examples of such observations:

All employees (even those not involved in SDLC) are required to complete basic SDLC training. The training includes various public and internal courses. The list of courses is expanded regularly.

Monthly security seminars are organized by the sales team to keep the awareness high.

No refresher is organized for the tech team as most of the employees are highly skilled security professionals. Regular pen tests and threat modeling workshops keep the people sharp.

The organization offers customized training for everyone who is part of team red and team blue.

For a more thorough sample set of observations we refer to the **TurboScale Solutions Case Study** from the SAMM Fundamentals Course.

Interview planning

An assessment interview requires careful planning.

Interview setup

Setting up the meetings

You should be selective in terms of who sets up the meetings. In some cultures, co-operation improves if the boss does that.

Interview format

We recommend planning 3 to 5 interviews of 1.5-2 hours each scoped per topic. SAMM's business functions are typically a good starting point for your interview topics.

List of stakeholders

Select relevant stakeholders for each interview. For instance, for the Governance business function interview you might need people involved in governance and security champions. For the Verification business function interview on the other hand you might need business analysts, application architects, developers, QA and/or project managers. Try to keep the interview group size small. Minimize the spectators as well. People will open up more in smaller groups.

Pre-interview kick-off briefing

It is a good idea to organize a prep kick-off briefing to let the interviewees know in advance of what's to come. Such as, the purpose of the interview, the format and the length of the sessions, which co-workers of the interviewees you work with, terms of confidentiality. If the interviewees are not familiar with SAMM it is also a good idea to provide at least a high-level overview of the model. To optimize the overall schedule we would recommend scheduling the pre-interview briefing immediately before the first session on Governance.

Live sessions vs online calls

Live sessions are preferable over conference calls. People will likely trust you more if they meet you in person. Hence they are more likely to open up. If for practical reasons live sessions are not feasible, make sure to always have your camera on even if the interviewees do not follow. That will help you gain some trust.

Interview preparation

From your end you should consider working with two people, namely the interviewer and the note taker. Study the organization roughly if you are not familiar with it yet. You might consider asking them to provide you with any relevant documentation in advance. Typical documents that are relevant in this context include amongst others organizational policies and standards, process-related documents, artifacts from completed activities, etc. We recommend you write an interview guide with open-ended questions based on the information you need to obtain (SAMM questions in this case). The guide provides the structure of a conversation instead of a long list of questions. We have provided a sample interview guide in the appendix. Finally, make sure to book time after the interview to consolidate your notes.

Interview questions

From your end you should consider working with two people, namely the interviewer and the note taker. Study the organization roughly if you are not familiar with it yet. You might consider asking them to provide you with any relevant documentation in advance. Typical documents that are relevant in this context include amongst others organizational policies and standards, process-related documents, artifacts from completed activities, etc. We recommend you write an interview guide with open-ended questions based on the information you need to obtain (SAMM questions in this case). The guide provides the structure of a conversation instead of a long list of questions. We have provided a sample interview guide in the appendix. Finally, make sure to book time after the interview to consolidate your notes.

Focus on the actuality

When conducting the interviews try to focus on what we call the actuality. Always ask how things have been going and not how things should be. For instance, “when was the organizational policy document last updated” instead of “is the organizational policy updated frequently”.

Ask open-ended questions

Open-ended questions are ideally not copies of SAMM questions, but meant to get the interviewee to talk on subjects in which the SAMM questions are likely to be answered. During that answering process, you can further help by mentioning topics that the interviewee doesn’t come up with. For instance, if the conversation is about organizational policies you could ask “and how about compliance obligations, are they relevant?”.

The relevance and importance aspect

In general, it is a good idea to question how relevant and important a certain activity is. For instance, some teams might not find incident detection relevant as they mostly work on proof-of-concept solutions. Questioning the importance of various security practices could also provide some additional information for the improvement roadmap planning phase.

Be a detective

Avoid communicating with questions about what is right or wrong. Try to find out what are the organizational realities as someone who is simply curious to understand them. For instance, instead of asking “Do you use checklists during threat modeling”, it is better to ask “Describe the threat modeling process” and then listen to see if checklists are mentioned.

The ‘rate’ trick

A good strategy to get people to open up and get them going is to ask them to rate certain things on a scale of 1 to 10 and then ask why.

Focus on feelings

Asking people about their feelings rather than just their thoughts can be a more effective way to encourage them to open up. This approach can foster a deeper level of communication and understanding. For instance, you could ask the interviewee “How do you feel about the added value of your threat modeling process?“.

The interview process

Starting the interview

Ask whether you may record all the interviews. In our experience the recordings (and their transcripts) can be extremely helpful during the note consolidation process. Beware that in some cultures it works better not to record the interview. Even if people allow it, the tendency to speak the truth is less and people feel less comfortable. If you are going to record, make sure that you clearly mention that and schedule an automated destruction of the recordings after a certain period of time (e.g., 30 days).

Break the ice

Invest some time in the interviewees even if it’s just to chitchat to lighten up the atmosphere. If you do a round of introductions, you can start for example and make your introduction a bit more personal, to invite others to do the same.

It is not an exam

Be courteous, friendly, respectful and humble. Try to connect and always remember that the interview is a collaboration, not an interrogation. You are not an auditor, but rather you want to assess a situation the way it is, so you can provide valuable insights on how to improve.

Be supportive

Appreciate that people may feel proud or threatened. Try to avoid negativity or judgment in your questions and reactions. Listen and encourage responses with enthusiasm, such as, “I see”, “That makes sense”, “Given your risk profile that is a reasonable strategy”. Paraphrase and ask follow up questions. Be curious.

Ask for artifacts

It is a good idea to ask for evidence for some of the answers. If you do this early in the session people will realize that you might check their answers and try to be more truthful.

The power of silence

Aside from listening and not interrupting it is a good idea to be silent for a while. Long silences can help people to open up.

Keep a natural conversation flow

Allow the conversation to flow in a natural way. This means that some of the questions you prepared may be already answered in the natural conversation. It also means that you need to pick your questions based on the topic where the conversation is. Hence, not necessarily the topic that was next on your question list. This requires solid focus and a good bookkeeping of which questions and quality criteria have been covered. Your note-taking colleague should be able to help with this.

Keep it structured

Give interviewees a sense of structure during the interview by providing transitions between major topics, at a quiet moment in the dialogue. For example “Now that we have discussed X, let’s move to Y”. It will help people focus, and it will give them confidence in your approach. Counter-intuitively, it is better to avoid sharing a precise structure at the beginning of the interview, because you don’t want to restrict people in discussing topics. Natural conversations tend to go back and forth, and it is better to let it flow like that without controlling it too much.

Note taking

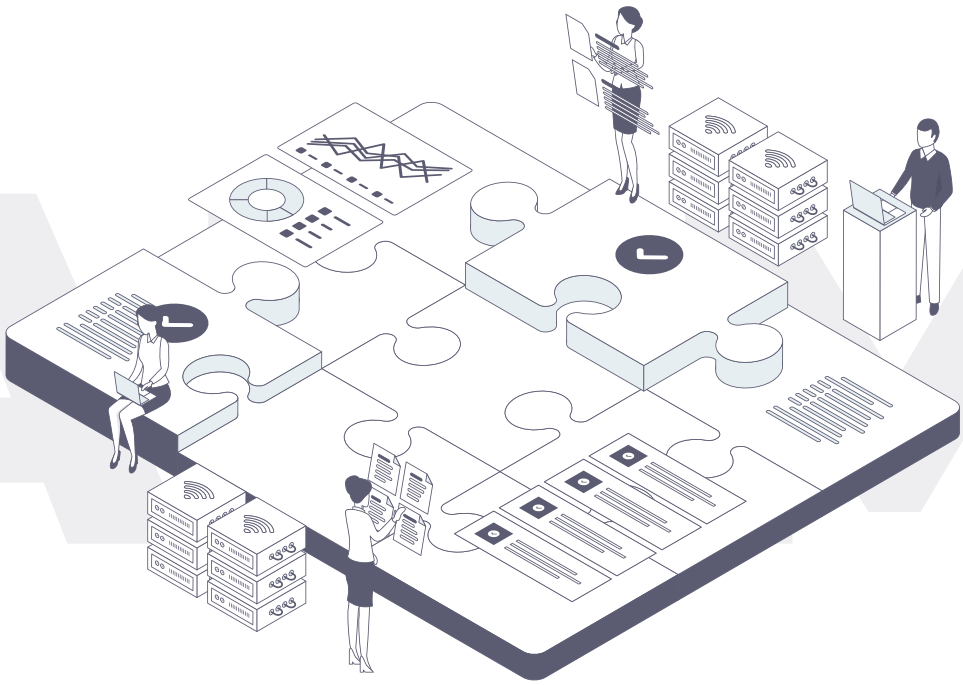
A we have already mentioned the note taker notes everything. Use verbatim unless completely sure of interpretation. For online sessions you could also leverage online transcription. In that case the note taker can be processing the conversations into more structured and interpreted notes. If you do the interview online, it is a good idea to keep a backchannel communication with your note taker (e.g., via Slack). Do make sure though not to cause any distraction by the note taking process itself. Don’t bring one of those loud keyboards or make use of the mute feature in online calling tools.

The end of the interview

Consider ending with ‘Is there anything you think we forgot to discuss?’
Finally, consider asking any sensitive questions after the interview is officially done.

Post-interview validation

It is good practice to present the preliminary assessment result (final answers to the questions) to the assessed organization in order to validate the outcome as well as to check whether the organization has any suggestions. To make this process effective, make sure the assessment rationalizes every decision.



Conclusion

This whitepaper provides a structured approach to leveraging the Software Assurance Maturity Model (SAMM) for improving software security. By utilizing the SAMM framework, organizations can effectively assess their current security posture, identify areas for enhancement, and develop a strategic improvement roadmap. The outlined best practices, tools, and methodologies enable a thorough and accurate evaluation, guiding organizations towards a more secure and resilient software development process. Implementing SAMM assessments will help organizations achieve and maintain robust software security, ensuring long-term protection and compliance.
For organizations looking to try out the SAMMY tool, we encourage you to get in touch with Codific. Additionally, Toreon offers expert coaching to help accelerate SAMM adoption and increase its impact in a scalable way. Partnering with these experts can provide the necessary support and guidance to optimize your software security practices effectively.

Appendix A: Sample interview questions

In this appendix, we present open-ended questions designed to trigger conversations that cover the range of SAMM questions as much as possible. You need to keep the SAMM assessment sheet at the side as a checklist, to see if you need to ask additional questions.

The scenario for these questions is an interview with a development team. The questions follow the order of the SAMM assessment sheet 2.0, but with a slight change for a more natural flow. Concretely, Education & Guidance comes after Secure Architecture.

1. Tell us about the input instructions, policies, standards or any other documentation that you get from the organization regarding security?

a. How does the organization deal with risks and threats? What assets and data are important and how?

b. How is that security documentation accessible? Is there anything else information-wise that you might need and that is not currently available?

c. What is the organization's strategic plan for dealing with application security? More specifically, you can ask about concepts like risk appetite, business priorities, roadmap, budgets, etc? Is there a buy-in from all stakeholders including the teams?

i. When was the last time the plan was reviewed and updated?

ii. How do you know it is a good plan? Are there any KPIs security-wise? (This question tries to look for metrics, how they are measured and where they are published. Furthermore the strategy and the roadmap are hopefully based on the KPIs and updated based on the results).

d. How about organizational policies and standards?

i. Describe how you interpret / implement these from a specific technology perspective.

ii. How do you ensure the organizational policies and standards are actually implemented? (This question is looking for test scripts / run-books / tools either manual or automated)

iii. Do you need to report back whether the applications are compliant with these policies and standards? How do you do that?

iv. Do you have any external obligations (e.g., GDPR, ISO27001, SOC2, etc)? How do you deal with these? (This question checks for any translation of these to requirements).

e. How does the organization deal with the inherent business risks of an application? For example, an intranet-facing app for calculating the holidays is obviously not as critical as a SaaS application that generates a substantial portion of the organization's revenue.

i. Is there any documentation of this risk profile per application?

ii. Does the team know about these and can they access these profiles?

iii. When was the last time it was updated?

2. Moving on to technical risks, how do you deal with risks and threats in a given application?

a. What approach do you use? Is there a checklist-based threat modeling approach or just a brainstorming session?

b. Is there training for such a risk/threat modeling activity?

c. What is the output of this exercise?

d. When do you threat model in your SDLC?

e. When was the last time you reviewed a threat model?

3. How do you deal with requirements in general and security requirements in specific? Do teams come up with requirements themselves and based on what input?

a. Could you please show some examples?

b. We're curious to understand who is involved in the requirements process?

c. Are the requirements aligned with the organization baseline (e.g., a password should be 12 characters long and not be found in a dictionary attack)?

d. Any chance you leverage a standard framework to assist (e.g., ASVS)?

4. Do you have any third parties building software for you? If yes, what do you require of them?

a. Do you have a standard list of requirements / questions for selecting the third party?

b. What does the agreement with the third party look like? Is there anything about security requirements and processes in that agreement?

c. Do you base your agreements on a standard template?

d. How do you check the security quality of the software the third party has delivered?

e. On a scale of 1 to 10 how good is the secure SDLC of this third party?

5. How do you deal with security requirements on an architectural level? Do you have any standard and reusable security building blocks or services?

a. Do you have a list of high-level security principles that the team knows and leverages (e.g., defense in depth, no secret sauce, etc.)

b. When you develop new applications, how do you build the architecture? Do you start from a reference architecture or are teams improvising every time?

c. Is there a list of available building blocks / services that you can simply reuse in every application (e.g., authentication module / service, sanitization service, etc). Also, do you have the necessary knowledge accessible to use these (e.g., if you have to plug in the sanitization service, is there someone you can ask for help who knows it)?

6. What technologies do you use? How about the security risks of those technologies (e.g., barebone frameworkless PHP applications like WordPress are horribly insecure)?

a. Can dev teams use any technology they would like to?

b. IIs there any automated enforcement of the recommended list of technologies if any?

7. How do you make sure that everyone is aware of security in terms of training?

a. What training do you provide to whom and when was the last time the training curriculum was updated?

b. Is the training for everyone? Is it consistent and mandatory?

c. Is there any customization towards different roles?

d. Is there a learning management system to track the completion of the courses?

8. Do you have security champions in the teams?

a. What do they do?

b. Is the process smooth and what would you change?

9. Do you have a security team / center of excellence?

a. What is its charter and what do they do?

b. Is there any material they publish and how is it accessible / searchable for everyone in the organization?

c. Is the security team effective and how could it be further improved?

10. How do teams share AppSec information?

a. Is there a portal / slack channel? How do you make sure people actually read it (especially in large organizations things might be communicated with massive amounts of emails ending up in nobody reading them)?

b. What information is shared? Tool updates? Standard changes? Metrics?

c. How well does it work on a scale from 1 to 10 and how would you make it better?

11. Could you describe your build process?

a. How do you harden / secure the process and the tools? Can anyone tamper with them?

b. How do you store the build-time secrets?

c. How much human interaction is required in the build process?

d. Do you have any security checks in the build? What tools do you leverage? Are the tools effective?

e. Do the tools break the build when they find any vulnerabilities? When do you accept a build and when not?

f. Do you log the warnings and failures?

g. How often do you reconfigure and reconsider the tools?

12. How do you deal with third party dependencies?

a. How do you deal with vulnerable dependencies?

b. How about outdated or no longer maintained dependencies?

c. How about the legal (licensing) aspect?

d. What happens to the build process if there is a vulnerability found in one of the third party libraries?

e. Which tools do you leverage for third-party dependency monitoring?

f. When a developer decides to add a new dependency, is there any approval process behind it?

13. Could you describe your deployment process?

a. How is the process hardened?

b. How much human interaction is required in the deployment process?

c. What happens if vulnerabilities are identified during the deployment?

d. How do you make sure that what you deploy is actually what you have produced in your build process (integrity of the deployed artifacts)?

e. How do you deal with production secrets? Who has access to them and is there any monitoring for suspicious events / access?

f. Can the source code end up containing secrets?

g. How do your secrets end up in configuration files?

h. How frequently are the secrets changed?

14. How do you track defects in your landscape?

a. Do you have a registry for it?
b. How do you classify your known weaknesses in type and in severity?
c. What are the sources?
d. Is there an SLA on these defects and do you have KPIs/metrics?
e. How is follow-up on defects enforced?

15. How and when do you review your architecture for security?

a. Who does it and how?
b. Do you have any standardized security requirements for that?
c. Do you track the findings?

d. Do you approach it from the perspective of standard threats, or maybe threats identified from threat assessment?
e. Do you also review the effectiveness of controls in that architecture? How?
f. Do you update your reference architecture based on findings across other development practices?

16. What is your security testing policy?

a. When are tests executed?
b. What part of the tests is automated and what part is not? For instance, how does the QA know what to test?
c. What frameworks / tools / languages do you leverage (test suits, DSLs, DAST, SAST, fuzzing)?
d. When do you translate bugs into regression tests?
e. Do you have a set of predetermined threats (abuse cases) to test against?
f. How about stress tests / denial of service?
g. How is security testing performed during build and deployment? What do you do with the findings?
h. How about manual review? Who, when and how do you perform it? Any checklists?
f. How about pentesting: when, how and who? Any specific test cases?
g. How do you use findings from testing to improve the development process / training?

17. Do you analyze logs for security incidents? What is the process of handling them (document, respond, root cause analysis)? Always ask how, by whom and when.

a. How formal are these detection and handling processes (owner, documented)?
b. When was the last time these processes were reviewed/updated?
c. How informed are people (training checklist)?
d. Do you have incident classification?
e. How about forensic analysis tooling?
f. Is there an incident response team?

18. How do you make sure your key techstack components are configured securely (e.g., container configurations, linux machines, etc).

a. Are there any configuration baselines in place and who maintains them?
b. Is there training for this?
c. How do you monitor conformity with baselines (automated)?

19. What is your process of keeping components up-to-date?

a. Do you have an SBOM and how do you check for vulnerabilities?
b. Is there an SLA in place?
c. How do you review and update this process?

20. How do you keep track of what type of data you process and where?

a. Do you leverage a data catalog?
b. Do you take into account any regulation?
c. How do you prevent production data getting into acceptance and testing? Who has access to the production data?

d. Are there any controls in place to protect data in its lifecycle?
e. How do you deal with backups and most importantly their timely deletion?
f. Do you audit and review the data catalog, policies and procedures?

21. What happens when applications or services become end-of-life?

a. How do you know which applications are still in use?
b. How do you review applications and services with respect to the life state and estimations?

Appendix B: Use SAMMY

The most efficient and scalable way to do SAMM assessments and to develop SAMM roadmaps is by using SAMMY. SAMMY exists in a freemium cloud version and an open source version.

You can find the freemium cloud version here: sammy.codific.com
The open source version here: github.com/codific/sammy

A guide on how to get started can be found here:
How to Get Started with OWASP SAMM on SAMMY

Appendix C: Get help

Contact Toreon for further assistance or to hire expert assessors.

toreon.com/contact-us/
Sebastien.Deleersnyder@toreon.com

TOREON
Business driven cyber consulting