CMP020X306 Secure Software Development: Group Portfolio

Threat modelling and managing security risks in a software development project

University of Roehampton

Set Date:	2 October 2024
Deadline:	**Five weeks** from the Set Date at 14:00 hours
Submission:	In-person: lab walk-through of Threat Model (Requirement 2) In-person: lab walk-through of Risk Treatment Plan (Requirement 3) In-person: lab walk-through of Software Development Plan (Requirement 4) In-person: lab walk-through of OWASP SAMM assessment (Requirement 5) In-person: 10-minute group presentation (Requirement 6) Moodle: Threat model with supporting diagrams (Requirement 2) Moodle: Risk treatment plan (Requirement 3) Moodle: Software development plan (Requirement 4) Moodle: OWASP SAMM assessment (Requirement 5) Moodle: Presentation materials (Requirement 6) Moodle: Individual security requirements (Requirement 7)
Feedback and Marks:	In-person and via Moodle
Marks (Group):	Maximum 34.0 marks for completion of requirements
Marks (Individual):	Maximum 10.0 marks for completion of requirements
Marks (Wow Factor):	Maximum 6.0 marks for Wow Factor
Requirements:	**As a group:** 1. Select **two** unique case studies with contrasting risk appetites. *For each case study:* 2. Propose a product architecture and perform threat modelling. 3. Use your threat model to create a risk treatment plan. 4. Write a software development plan based on the risk treatment plan. 5. Use the OWASP SAMM to assess the software development plan tool.

	Comparing and contrasting each case study: 6. Prepare and present a 10-minute group presentation of your work. **Individually:** 7. Elaborate three security requirements for each product.
Learning Outcomes:	**LO1.** Identify and manage security risks as part of a software development project **LO4.** Systematically develop and implement the skills required to be an effective member of a development team in a virtual professional environment, adopting real-life perspectives on team roles and organisation.

IMPORTANT: *This is a living document and will be subject to changes and updates during the life cycle of the lab portfolio. Therefore, it is imperative that you check this document regularly!!*

Academic Misconduct

Your submission for this coursework will be scrutinised for plagiarism, collusion, and other forms of academic misconduct.

Please ensure that the work that you submit is your own, and that you have cited and referenced appropriately, to avoid having to attend an academic misconduct hearing.

Contents

About this lab portfolio

The focus of this portfolio is learning how to identify and manage security risks within a software development project. It aims to introduce you to a wide range of software security concerns, and provide practice in team activities that can support designing and building secure software systems.

You will work in groups to develop the knowledge and skills used in software teams aiming to produce software that is secure by default and design. This includes:

- Threat modelling using the four-question framework and STRIDE.
- Treating threats according to different risk appetites.
- Proposing and assessing software development practices to support your risk treatment.

You will also individually work on security requirements, again, mirroring real-world work where your individual work fits within and contributes to a group activity.

There is a flow to the requirements which mirrors software development and the four-question threat modelling framework. Therefore, we recommend that you address the requirements in your lab time as follows:

- **Lab 1:** Requirements 1, 2 and 3.
- **Lab 2:** Requirements 4, 5 and 7.
- **Lab 3:** Requirements 6.

This portfolio lab is problem-centred. You will need to research and take responsibility for solving challenges that you encounter. This includes making appropriate assumptions if the case studies do not contain all of the information you might like.

As final-year coursework, it is likely that additional time will be required between lab sessions to complete the work to obtain higher grades.

You should work collaboratively with the other students in your group. However, the work that you submit must be your own (for individual elements) and your groups (for group elements). Any sources of assistance, including AI, must be acknowledged through referencing.

Use of AI tools

Different companies have different policies on the use of AI tools, as, amonst other things, they seek to balance their compliance, data protection and intellectual property obligations.

The use of AI tools can present software cyber security threats. For example:

- Export of code outside of the organisation, resulting in:
  - Third-parties identifying vulnerabilities.
  - Compromise of security tokens held in the code.
  - Data transfers across national boundaries with security or data protection implications (e.g. ITAR, GDPR).
- Generation of code with security vulnerabilities.

Within this portfolio, your group *may* make use of AI tools for completing work on **one** of the two case studies. However, you are required to disclose your use of them in your group's presentation. You may not submit work solely or significantly generated by AI tools.

Two examples of reasonable use of AI tools might be:

1. If you are having difficulties understanding the description of the product's architecture, using DiagramGPT may help produce a starter diagram to get you started with threat modelling.
2. If you are finding it challenging to think of threats, asking an LLM agent to suggest some might help you get started.

Marking Criteria

The **maximum mark** for completing the **group requirements is 30**, with **10 marks for your individual contributions**.

An **additional maximum mark of 10** can be awarded for "**Wow Factor**" that evidences appropriate, relevant and additional learning. Typically, wow factor demonstrates a self study contribution that extends or advances the core technical requirements of a lab portfolio.

To receive a mark for this portfolio lab, you need to meet the lab submission requirements, including the group presentation. The presentation needs to clearly evidence the requirements described in this document.

This portfolio will be marked in accordance with the following rubrics.

Marking Rubric

Marks will be awarded according to the following rubric:

Criteria	Excellent	Very Good	Good	Developing	Not Attempted
Case study selection Identifying case studies with significantly different risk appetites, agreeing why they were selected, and any ambiguities and assumptions.	Selection performed without support from outside the group. Excellent articulation of reasons for selection, any ambiguities and assumptions made.	Selection performed without support from outside the group. Good articulation of reasons for selection, any ambiguities and assumptions made.	Selection performed without support from outside the group. Satisfactory articulation of reasons for selection, any ambiguities and assumptions made, with some obvious errors, omissions or lack of detail.	Supported required from outside the group. Limited articulation of reasons for selection, any ambiguities and assumptions made, with multiple obvious errors, omissions or lack of detail.	Significant support required, or selection of case studies without contrasting risk appetites, failure to agree reasoning for selection.
	2 Marks	1.5 Marks	1 Marks	0.5 Marks	0 Marks
Threat modelling A structured process for identifying relevant threats and what you are going to do about them.	Excellent architecture diagram that clearly conveys system architecture. Systematic approach to threat modelling. Appropriate selection of data flow diagrams according to requirement. Effective use of OWASP Threat Dragon for data flow modelling clearly identifying assets, trust boundaries and data flows. Relevant threats are assigned to elements in each data flow diagram in Threat Dragon. Adoption of standard diagramming approaches e.g. DFD3, C4 model.	A near-completed threat model which does quite meet the requirement in some way. For example, it may be missing a data flow diagram. The threat modelling that has been done should have been completed to a very good standard in a systematic way. For example, a clear architecture diagram should be present, and effective use should have been made of OWASP Threat Dragon for data flow modelling, with relevant threats assigned to elements in each data flow diagram.	A threat model with approximately half of the required data flow diagrams completed. However, the threat modelling that has been done should have been completed to a good standard in a systematic way. For example, a clear architecture diagram should be present, and effective use should have been made of OWASP Threat Dragon for data flow modelling, with relevant threats assigned to elements in each data flow diagram.	A basic architecture diagram is present but too underdeveloped to be effective for supporting threat modelling. Some attempt at threat modelling has been performed, with data flow diagrams and threats in OWASP Threat Dragon, although these may be limited. Threats identified may not be particularly relevant to the case study. Approach taken may not be systematic and structured.	Failure to produce an architecture diagram. Failure to consider the development environment DFD. Little or no evidence of a systematic approach to threat modelling. Little or no use of OWASP Threat Dragon. Fewer than five threats identified, and those identified may not be relevant or appropriate.
	8 Marks	6 Marks	4 Marks	2 Marks	0 Marks

Criteria	Excellent	Very Good	Good	Developing	Not Attempted
Risk Treatment Plan Using the threat model to prioritise and appropriately mitigate the threats identified.	Excellent evidence of appropriate scoring of threats in the Threat Dragon threat model with a consistent approach taken which considers the organisation's risk appetite. Threats prioritised appropriately based on the scoring assigned. Appropriate security controls identified to mitigate the threat.	Very good evidence of appropriate scoring of threats in the Threat Dragon threat model with a consistent approach taken which considers the organisation's risk appetite, with perhaps one or two omissions. Threats prioritised appropriately based on the scoring assigned. Appropriate security controls identified to mitigate the threat.	Good evidence of scoring of threats in the Threat Dragon threat model with a consistent approach taken which considers the organisation's risk appetite, with some omissions, unscored or unmitigated threats. Threats prioritised appropriately based on the scoring assigned. Appropriate security controls identified to mitigate the threat.	Some scoring of threats in the Threat Dragon threat model. Approach taken may be inconsistent or conflicting with the organisation's risk appetite. Some security controls identified to mitigate the threat.	Little or no scoring of threats in the Threat Dragon threat model. Little or no security controls identified.
	8 Marks	6 Marks	4 Marks	2 Marks	0 Marks
Software Development Plan Develop a secure software development plan to address threats arising during development.	Excellent organisation, points are logically ordered and linked to threats identified in the threat model. Security properties during the development lifecycle are clearly communicated. Controls identified are appropriate for the organisation's risk appetite.	Clear organisation, points are logically ordered and linked to threats identified in the threat model. Security properties during the development lifecycle are clearly communicated. Controls identified are appropriate for the organisation's risk appetite with one or two areas below the standard expected.	Organised, but lacking clarity in places. Most activities are linked to threats identified in the threat model. Security properties during the development lifecycle are given some consideration. Most controls identified are appropriate for the organisation's risk appetite, but may signficiantly over- or under- apply security controls in some areas.	Some organisation. Some activities are linked to threats identified in the threat model. Little or no consideration of security properties during the development lifecycle. Controls may be inappropriate for the risk appetite, or wishful thinking. Would be challenging to apply in a software development environment.	Poorly organised. No linkage to threats in the threat model. No consideration of security properties in the development lifecycle. Little evidence of practical security controls for a development team to apply.
	4 Marks	3 Marks	2 Marks	1 Marks	0 Marks

Criteria	Excellent	Very Good	Good	Developing	Not Attempted
SAMM Assessment Using the OWASP SAMM assessment to assess the maturity level of the software development plan.	An excellent assessment of the software maturity level for each of the two case studies. Answers and commentary are professional and appropriate, with assumptions documented.	A very good assessment of the software maturity level for each of the two case studies. There may be small gaps in the assessment, or minor inconsistencies. Answers and commentary are professional and appropriate, with assumptions documented.	A good assessment of the software maturity level for each of the two case studies. There may be several gaps in the assessment, for example, where assumptions were not made to aid completing the form. Answers and commentary are professional and appropriate.	An attempt has been made to complete the assessment. Significant areas of the assessment remain incomplete. Assumptions made may be inappropriate.	Little or no evidence of the assessment having been completed. Answers may be inappropriate or unprofessional.
	4 Marks	3 Marks	2 Marks	1 Marks	0 Marks
Presentation Clearly communicating the group's work including threat modelling, treatments and SAMM assessments to influence others.	All of the group is present. A professional and compelling presentation that meets or exceeds the requirements and is likely to influence others. Clearly communicates the work done, the threats identified, mitigations and development security controls. Excellent use of supporting material such as diagrams, screenshots etc.	Some of the group may be missing. A professional and compelling presentation that meets the requirements, or which only falls short in very minor ways. Clearly communicates the work done, the threats identified, mitigations and development security controls. Very good use of supporting material such as diagrams, screenshots etc.	Some of the group may be missing. A professional presentation that meets the majority of the requirements. Clearly communicates the work done, the threats identified, mitigations and development security controls. Good use of supporting material such as diagrams, screenshots etc.	More than half of the group may be missing. A presentation that meets some of the requirements. Communicates the work done, the threats identified, mitigations and development security controls. Limited use of supporting material such as diagrams, screenshots etc.	More than half of the group may be missing. Little or no evidence of preparation. Presentation is unprofessional, inappropriate, or fails to address the requirements. Little or no evidence of the work done, threats identified, mitigations and development security controls.
	8 Marks	6 Marks	4 Marks	2 Marks	0 Marks

Criteria	Excellent	Very Good	Good	Developing	Not Attempted
(Individual) Security Requirements Translating security controls into specific requirements that a software development team could understand and deliver.	Excellent structure providing the required attributes and narrative for each of the two case studies. Selection of appropriate security requirements. Clearly articulates the work required with appropriate and easy to understand verification critieria. References to appropriate supporting information provided where useful, for example, to external standards. Appropriate and professional medium used.	A very good set of requirements that does not fully meet the requirement in a limited way. The work done should still be well organised and presented, with appropriate selection of requirements. The requirement should be clearly understandable, although there may be minor omissions in verification conditions.	A good set of requirements that meets the majority of the requirements. The work done should still be well organised and presented, with appropriate selection of requirements. The requirements may be less clearly communicated, or lack precision in verification conditions which would make completing the work harder and potentially require rework.	Security requirements have been identified, but have not been fully developed and the majority of the requirement is not met. Answers may be missing several required attributes, or fail to communicate appropriately the work required or the verification conditions. This would likely prevent a developer being able to start the work until the requirements were reworked.	Security requirements may have been identified, but likely only at a headline level. Little or no evidence of addressing the requirement, or providing sufficient detail for a developer to consider implemeting the requirement.
	8 Marks	6 Marks	4 Marks	2 Marks	0 Marks
(Individual) Submission The submission meets the requirements described.	Coursework has been submitted completely in a correctly-named Zip file and contains all the files necessary.	Coursework submission missing a single minor element (e.g. incorrectly named Zip).	Coursework submission missing two or more element, or missing files required to fully demonstrate completeness.	Coursework submission is missing several elements, but demonstrates singificant proportion of the work.	Coursework submission missing many elements and/or is not a professional submission.
	2 Marks	1.5 Marks	1 Marks	0.5 Marks	0 Marks
Wow Factor Demonstration of self-directed additional learning related to the portfolio.	Evidence of an excellent attempt that is relevant to the portfolio. Able to clearly explain the work done, value of it and relevance to portfolio.	Evidence of a very good attempt that is relevant to the portfolio. Able to clearly explain the work done, value of it and relevance to portfolio.	Evidence of a good attempt that is mostly relevant to the portfolio. Able to explain the work done, value of it and relevance to portfolio.	Evidence of an adequate attempt that is somewhat relevant to the portfolio. Some ability to explain the work done and relevance to portfolio.	No additional Wow Factor, or evidence of a very limited attempt that is not relevant to the portfolio.
	6 Marks	4.5 Marks	3 Marks	1.5 Marks	0 Marks

Late Portfolio Submissions

For each week that a portfolio is late, five marks will be deducted from the portfolio score that is awarded.

Resources Required for this Portfolio Lab

You may find the following course resources useful:

1. Software: OWASP Threat Dragon
2. Reference: OWASP Threat Modelling Cheatsheet
3. Reference: OWASP Application Security Verification Standard
4. Reference: OWASP SAMM assessment spreadsheet
5. Reference: OWASP Software Assurance Maturity Model (SAMM)
6. Reference: OWASP Software Assurance Maturity Model (SAMM) (PDF)

Requirements

Requirement 1. Select two unique case studies with contrasting risk appetites

Different organisations have different views on how much risk is acceptable. This can be driven by different markets, different legislation or regulatory regimes, different business lifecycle stages (start-up vs established business) and different people. We refer to the level of risk an organisation is prepared to accept as the *risk appetite*.

We have generated a range of fictional case studies using GenAI Large Language Models (LLMs). Each case study provides:

- The company name, company profile and a description of their risk appetite.
- Information on a software product that your group is responsible for.
- Information on the product's architecture, users and data.
- A brief indication of employees' awareness of cyber security.

Your group is required to select **two** case studies from those available. These should be distinct from those selected by other groups.

The case studies your group selects **must have significantly different risk appetites**. This is important as you will need to compare and contrast how you are treating risks differently in the different organisations.

Your group may choose case studies that are similar or different in other ways (e.g. similar product or architecture) as long as the risk appetites are significantly different.

In addition to selecting the case studies, your group should:

1. Agree on why you have selected these case studies given the other activities in this portfolio.
2. Identify areas of ambiguity in the case study, and agree and write down any assumptions you are making to clarify them.

Requirement 2. Propose a product architecture and perform threat modelling

For each case study, your group is required to:

1. Create a high-level architecture diagram for the product in the case study.
2. Agree the Data Flows Diagrams (DFDs) you will create.
3. Create a new Threat Model in OWASP Threat Dragon, completing the model information appropriately.
4. Develop the agreed dataflow diagrams in OWASP Threat Dragon.
5. Export Threat Dragon's Report as a PDF, and save the Threat Model in JSON format. Ensure you save your threat model regularly during development. You may wish to use version control (e.g. GitHub) to ensure that you do not lose you work, and that you can share it easily amongst your group.

You may find it quickest to work on a whiteboard or paper first, then transfer the work into software.

Architecture diagram The purpose of this is to quickly create a shared understanding of the system. You will use this understanding to help create your threat model.

There is no requirement to use a particularly diagramming approach or tool for creating your architecture diagram. However, you may like to use a recognised approach such as the C4 model or a simple box-diagram approach.

It is not important which tool(s) you use for this. You may use a whiteboard, pencil and paper, a computer diagramming tool, or a selection of these to come up with an agreed architecture based on your case study.

What is important is that the software architecture is based on the description in the case study and clearly shows:

- Users;
- Software components and key processes;
- Connectivity, for example, computer networks;
- Datastores, including any queues and databases used;
- Any third-parties, for example, cloud providers or partner APIs.

You must ensure you capture this diagram electronically. For example, in a diagramming tool, or by taking photos of whiteboard or paper copies.

This diagram does not need to contain huge levels of detail, and you will not be required to build a system from this diagram.

Threat modelling

Four-question framework To guide your threat modelling, your group should use the four-question framework from the Threat Modelling Manifesto:

1. What are we working on?
2. What can go wrong?
3. What are we going to do about it?
4. Did we do a good enough job?

STRIDE Security threats should be identified using the STRIDE model for identifying security threats:

- **S**poofing
- **T**ampering
- **R**epudiation
- **I**nformation disclosure (privacy breach or data leak)
- **D**enial of service
- **E**levation of authority

OWASP Threat Dragon OWASP Threat Dragon is a modeling tool used to create threat model diagrams as part of a secure development lifecycle. Threat Dragon follows the values and principles of the threat modeling manifesto. It can be used to record possible threats and decide on their mitigations, as well as giving a visual indication of the threat model components and threat surfaces. Threat Dragon runs either as a web application or as a desktop application.

You may use either the publically-available instance of Threat Dragon at https://www.threatdragon.com/ or install your own local version using the instructions on the OWASP page.

Threat Model, Data Flow Diagrams (DFDs) and Report Your Threat Model should include DFDs covering:

- A high-level overview of the entire system;
- Between two and five more focused DFDs which detail sub-systems or specific use cases.
- The development environment, which might include:
  - Version control;
  - Build and CI/CD servers;
  - Third-party service such as test runners or SAST tooling;
  - Datasources;
  - Secrets (e.g. passwords, deployment keys).

Each Data Flow Diagram (DFD) within your threat model should:

- Clearly identify the assets you are seeking to protect.
- Make appropriate and consistent use of DFD notation.
- Show trust boundaries and relevant data flows crossing them.
- Make use of per-element properties within Threat Dragon, for example, descriptions and identifying whether a datastore or data flow is encrypted.

Threats recorded against each element or data flow must:

- Be mapped to STRIDE;
- Have a clear description.

Once your risk treatment plan is completed, the threats must:

- Be appropriately prioritised after you have completed your risk treatment plan.
- If they are to be treated, have proposed mitigations or controls listed against them.

Note: Threat Dragon's Report is generated from the main screen once your threat model is loaded:

Requirement 3. Use your threat model to create a risk treatment plan

For each case study your group is required to use the threat modelling report created for Requirement 2 to agree a Risk Treatment Plan based upon the case study's risk appetite.

You will record the Risk Treatment Plan by updating your threat model so that the threats identified have:

- An appropriate score assigned as agreed by the group. This should follow a consistent approach, for example:
  - Agreeing values based on business impact;
  - Agreeing values based on likelyhood and severity and multiplying them together to get a single number.
- An appropriate priority assigned (Low, Medium, High) based on the scoring assigned.
- Appropriate mitigations identified against the threat.

Rather than reinventing the wheel, standard security requirements can help. You may find it helpful to consult the OWASP Application Security Verification Standard to help identify mitigations or controls to apply.

Moodle submission Your Moodle submission must include the following **for each case study**:

1. The architecture diagram in either PDF or JPEG format.

Threat Dragon v2.2.0-latest

Demo Threat Model

Owner:
Mike Goodwin

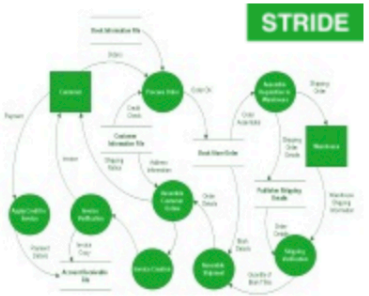
Reviewer:
Jane Smith

Contributors:
Tom Brown,
Albert
Moneypenny

High level system description

A sample model of a web application, with a queue-decoupled background process.

Main Request Data Flow



Edit Report Close Model

Figure 1: OWASP Threat Dragon's Threat Model main screen

2. The completed OWASP Threat Dragon **Threat Model** in JSON format.
3. The OWASP Threat Dragon *Threat Model Report* as a PDF.

We suggest the following filesystem layout, within a ZIP archive:

```
- Case study 1
  - /case_study.pdf
  - /architecture_diagram.pdf
  - /threat_model.json
  - /threat_model.pdf
  - /threat_model_report.pdf
- Case study 2
  - /case_study.pdf
  - /architecture_diagram.jpeg
  - /threat_model.json
  - /threat_model_report.pdf
```

Requirement 4. Write a software development plan based on the risk treatment plan

Many organisations will create some form of software development plan. Whilst some produce onerous documents, others may create simple markdown documents addressing the issues using bullet points.

For each case study you should create part of a software development plan addressing secure software development concerns, answering the question:

> What will you do to control the software security risks?

This should be based on the threat model and risk treatment plan you created for the development environment. It should include:

- Identifying security properties in the development lifecycle, including:
  - Confidentiality;
  - Integrity;
  - Availability.
- How you will secure your development environment?
- The use of appropriate tools that help with securing software development.
- References to the threat identifiers in the threat model to show where mitigations and controls are applied.
- Development practices that ensure the system is secure by design and secure by default.
- Security management, including:
  - Management of any access controls and secrets;
  - Preventing vulnerabilities;
  - Responding to vulnerabilities.

This should be no more than **four sides of A4** long.

Requirement 5. Use the OWASP SAMM to assess the software development plan

OWASP's Software Assurance Maturity Model (SAMM) is a measurable way for organizations to analyze and improve their software security posture.

The SAMM defines five business functions with three security practices in each function. These are assessed using three maturity levels.

You can use the SAMM to:

- Assess an organization's current software security posture;
- Define the organization's target software security posture;
- Define an implementation roadmap to go from current to target;
- Provide advice on how to implement particular activities.

More information on the SAMM and the assessment criteria are available on OWASP's Software Assurance Maturity Model (SAMM) page. There is also a PDF version of the SAMM.

For this requirement, you must:

1. Complete the SAMM Assessment spreadsheet for your each study's software development plan.
2. Make use of the *Interview Notes* field to record appropriate comments and references that support your assessment.

You are unlikely to have considered all of the areas in your software development plan. For some areas, in particular, the *Governance* function, you will need to make reasonable assumptions based on the information given in the case study. Document this in the *Interview Notes* field.

Requirement 6. Prepare and present a 10-minute group presentation of your work

Your group presentation should:

- Describe the case studies and why your group selected them.
- Describe the threat models you have created, showing the use of the four-question framework.
- Show how you scored and prioritised the threats identified to come up with your risk treatment plan.
- Evidence how the risk treatments are applied within your software development plans.
- Contrast the case studies' different risk appetites and how this reflects your risk treatment, security controls and software development plans.
- Compare and contrast the two different SAMM assessments, indicating where key differences arise, and where you have had to make assumptions.
- Summarise the different security requirements written, highlighing similarities and differences.
- Demonstrate all requirements for each case study were met, using screenshots, diagrams etc.
- Demonstrates awareness of a range of different security controls throughout the development lifecycle.
- Reflect upon your group's work, specifically the last question in the four-question framework: "Did we do a good enough job?"
- Indicate where you used any AI tools, which tools you used, and whether you felt they helped or hindered your work.
- Make effective and professional use of presentation slides.

This is a lot of material to cover in a short presentation. Your group will need to use the time effectively to showcase their work throughout the portfolio.

Requirement 7. Elaborate three security requirements for each product

As an individual you have now been assigned the Product Owner role. Within the case study organsation you are responsible for gathering and defining the product's requirements, and maximising delivered business value.

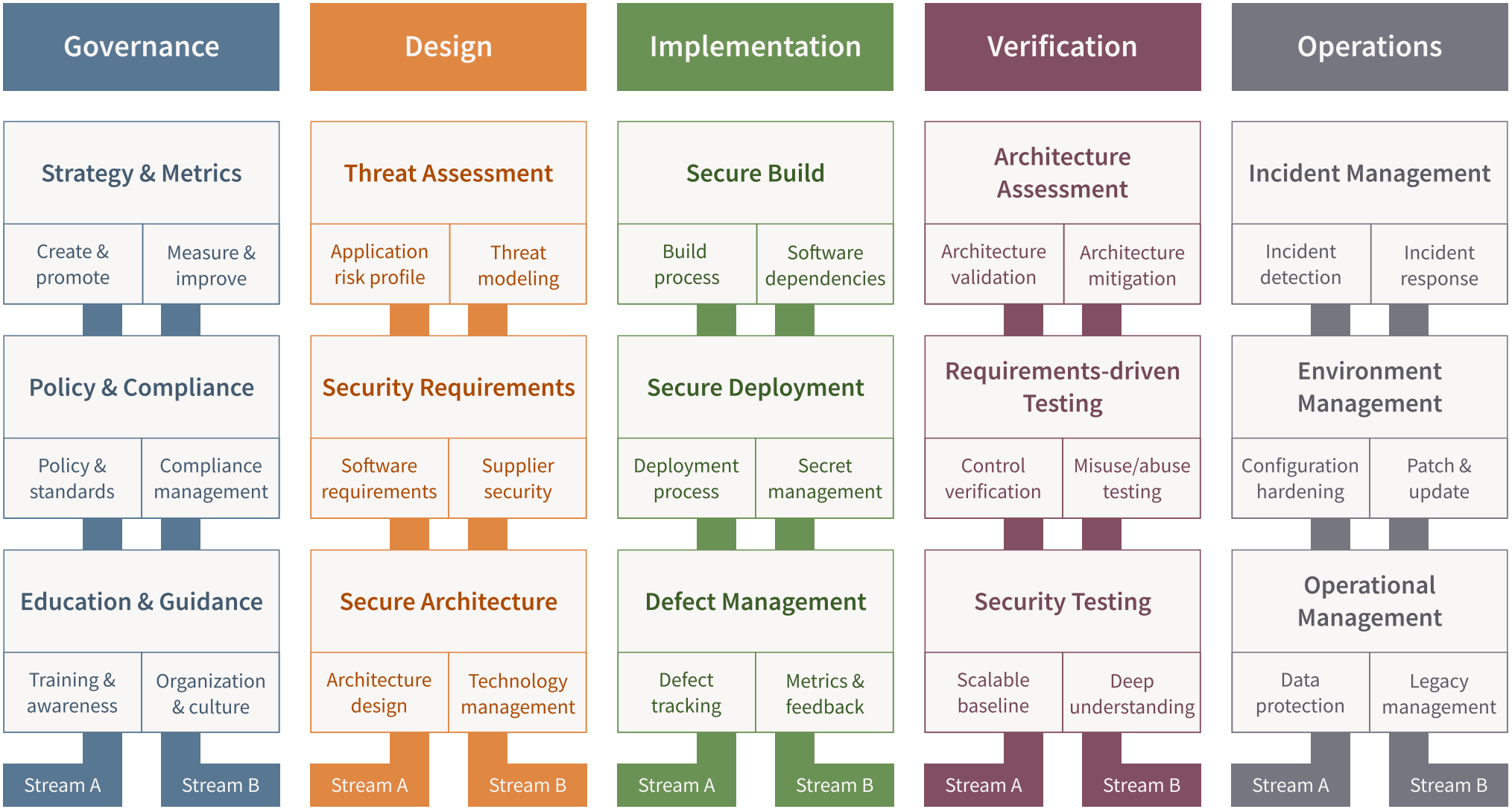


Figure 2: OWASP SAMM v2

Through Threat Modelling you have identified and prioritised a number of security requirements for the software team to deliver. However, to ensure they are delivered successfully you need to clearly communicate them as security requirements.

For each case study you should select **three** different threats that your group has chosen to mitigate and write a detailed security requirement for one of the mitigations or controls recommended.

As a group you must ensure that **each member of the group works on different security requirements**. Writing requirements for different mitigations or controls for the same threat is acceptable, as this provides *defence in depth* (DiD). For example, one person might write requirements covering code-level controls preventing injection vulnerabilities, and another might write infrastructure requirements for controls such as web application firewalls (WAFs) mitigating the same threat.

For each security requirement you are expected to provide:

- **Title:** A title for the requirement.
- **Asset:** Which asset are you protecting?
- **Threat:** Which STRIDE threat is this protecting against?
- **Narrative:** Describe the threat in context so a developer can understand it.
- **Work required:** Describe the work required to be done by the software team, for example, as a series of use cases or misuse cases.
- **Verification criteria:** How should the work be tested to show the requirement is implemented correctly.

This might be delivered as slides, a spreadsheet, or a written document.

Wow factor

It is feasible to pass this portfolio without completing any *"wow factor"*. However, if you decide to take on this additional learning opportunity, the choice of what to contribute is yours. Here are some examples to consider:

- Performing a 'deep-dive' on one specific area of a case study for more detailed threat modelling.
- Researching sector-specific security standards and including them in the software development plan or individual security requirements.
- Researching real-life cyber security incidents relevant to your case study, as part of building an evidence base for:
  - Why threats you have identified are appropriate and realistic.
  - Why the risk treatments identified are appropriate.
- Identifying specific tooling that addresses identified security risks, or may help a development team's security and productivity, for example:
  - SAST tooling.
  - Signed commits or software signing.
  - Dependency monitoring.
- Exploring an alternative threat modelling tool, for example, repeating a DFD with Microsoft Threat Modelling Tool.
- Exploring an alternative threats modelling approach, such as PASTA.
- Exploring automated parsing and processing of the threat model in a way that helps your group.

Demonstration Tasks

To receive a mark for this work, your group must demonstrate the extent to which they have completed the requirements and specifications of this portfolio, to your instructor.

You must demonstrate Requirements 2, 3, 4, 5, and 6 to your instructor.

You must also upload the following to Moodle as a Zip file:

- Threat model with supporting diagrams (Requirement 2)
- Risk treatment plan (Requirement 3)
- Software development plan (Requirement 4)
- OWASP SAMM assessment (Requirement 5)
- Presentation materials (Requirement 6)
- Individual security requirements (Requirement 7)

Your Zip file must use your student ID as a name. For example, if your student ID is abc1234, your Zip file should be called abc1234.zip

You will be individually marked on your security requirements (Requirement 7) once these have been uploaded to Moodle, with any *wow factor* taken into account.

NOTE: To ensure that instructor assessment time fairly distributed, each group is permitted **one formal demonstration period in each lab session**. After this, marks and an outcome will be recorded.

End of Portfolio :-)