Insecure Design

Alexander Tushinsky

Cybersecurity & Software Development Consultant

@ltmodcs alextushinsky.com

Overview

- What is Insecure Design?
- OWASP Software Assurance Maturity Model
- Threat Modeling
- OWASP projects that help to mitigate

Insecure Design

- Broad category that represents various vulnerabilities as missing or ineffectual
- Not the source for all other Top 10 risks
- Insecure design cannot be fixed by a perfect implementation

Preventing Insecure Design

Secure SDLC

Reusable Secure Component Library

Threat Modeling

Incorporate security into user stories

Unit and integration testing for all critical flows

Security at each tier of the application

How Does OWASP Help?

- OWASP Top 10
- OWASP Proactive Controls
- OWASP ASVS
- OWASP SAMM

OWASP SAMM

Software Assurance Maturity Model

Measurable
Defined maturity standards

Actionable
Methods for improvement

Versatile
Works across the organization

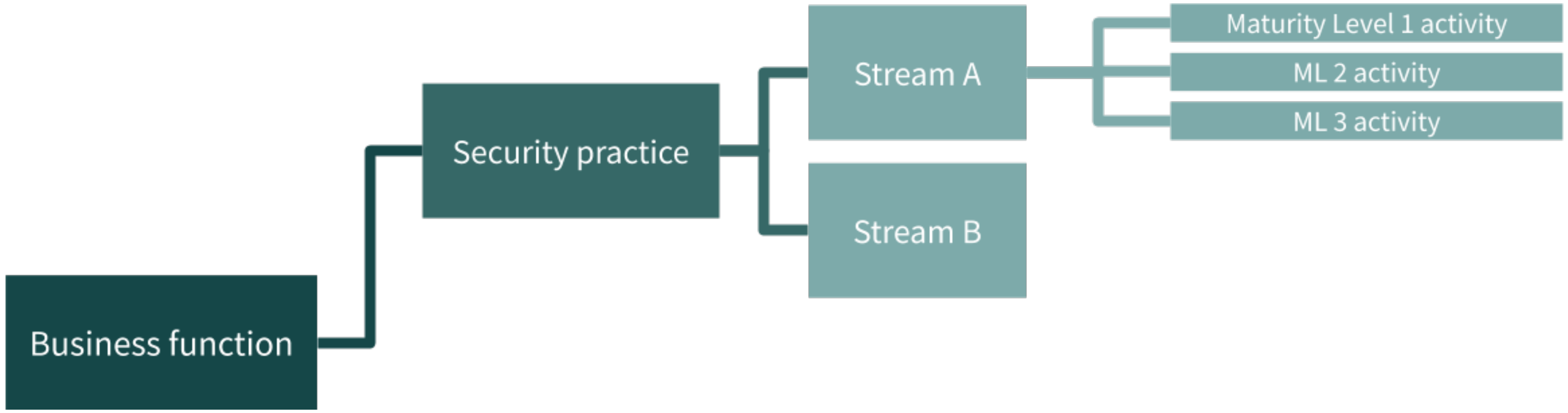
Business Function

Three Security Practices

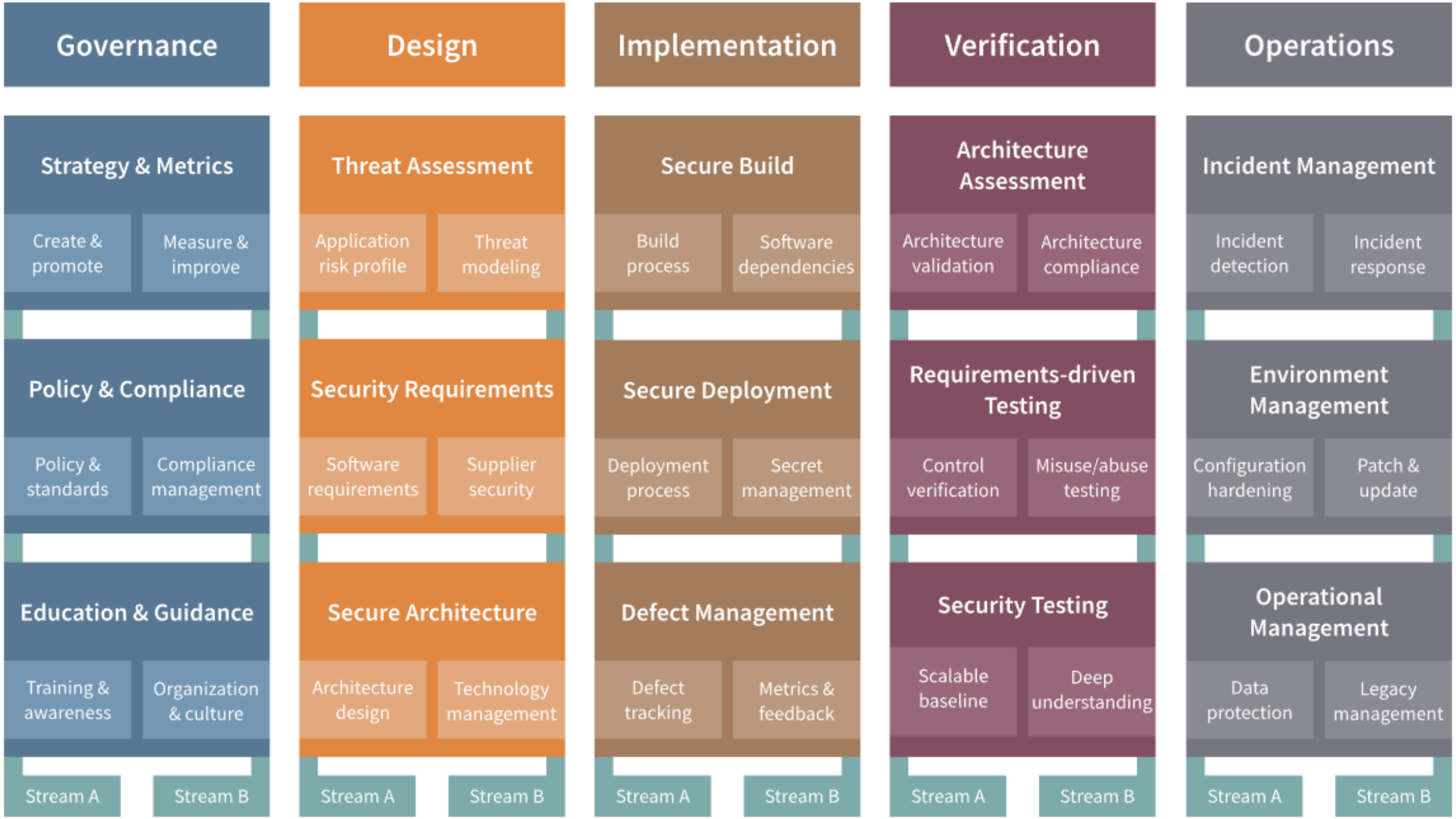
Activities divided into two streams

Three maturity levels per stream

Citation: https://owaspsamm.org/about/



SAMM



Governance	Design	Implementation	Verification	Operations
Strategy & Metrics	Threat Assessment	Secure Build	Architecture Assessment	Incident Management
Create & promote Measure & improve	Application risk profile Threat modeling	Build process Software dependencies	Architecture validation Architecture compliance	Incident detection Incident response
Policy & Compliance	Security Requirements	Secure Deployment	Requirements-driven Testing	Environment Management
Policy & standards Compliance management	Software requirements Supplier security	Deployment process Secret management	Control verification Misuse/abuse testing	Configuration hardening Patch & update
Education & Guidance	Secure Architecture	Defect Management	Security Testing	Operational Management
Training & awareness Organization & culture	Architecture design Technology management	Defect tracking Metrics & feedback	Scalable baseline Deep understanding	Data protection Legacy management
Stream A Stream B	Stream A Stream B	Stream A Stream B	Stream A Stream B	Stream A Stream B

Citation: https://owaspsamm.org/about/

Implementation

PREPARE

ASSESS

SET THE TARGET

DEFINE THE PLAN

IMPLEMENT

ROLL OUT

Threat Modeling

Threat Modeling Steps

Identify business goals

Diagram the application

Model the system

Identify threat agents

Use risk framework to rank threats

Identify how to mitigate threats

STRIDE

- **S**poofing – Can an attacker impersonate a legitimate user?
- **T**ampering – What can an attacker alter?
- **R**epudiation – Can we identify the attacker?
- **I**nformation Disclosure – Can a user see someone else’s data?
- **D**enial of Service – Can an attacker shut down our system?
- **E**levation of Privilege – Can an attacker gain additional permissions?

DREAD

- **D**amage – How bad would an attack be?
- **R**eproducibility – How easy is it to reproduce?
- **E**xploitability – How easy is it to launch the attack?
- **A**ffected users – Who is impacted?
- **D**iscoverability – How easily is this threat found?

DREAD

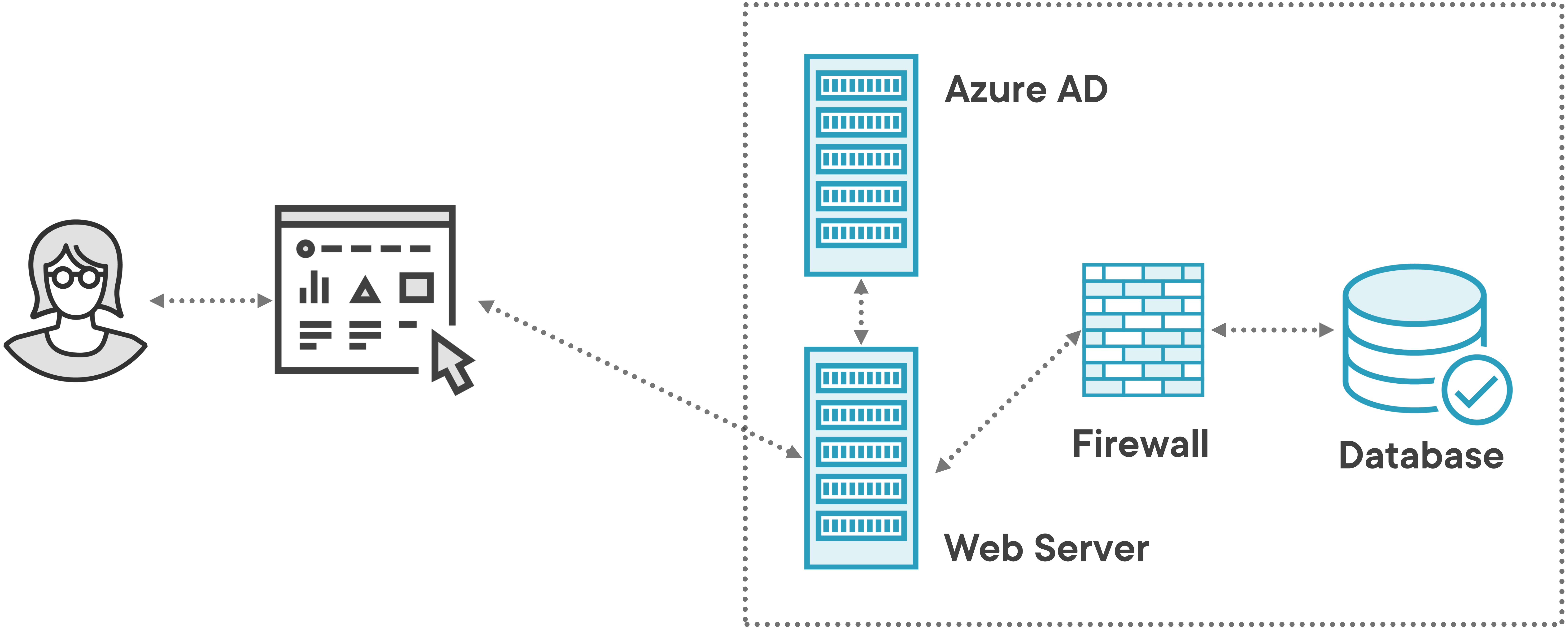
- Assign a score to each item:
  - **1** – Low, **2** – Medium, **3** - High
- Sum up to get the DREAD score:
  - **High** – 12 – 15
  - **Medium** – 8 – 11
  - **Low** – 5 to 7

Threat Modeling Examples

Simple Example

As a user, I want to login into the system so I can check my account.

Basic Diagram



Logon Flow Threats

S – Lack of 2FA allows an attacker to log in as a valid user.

T – Cross-site scripting attack may expose the cookie authentication, leading to spoofing.

R – IP address is not being logged, so traceability may be impossible.

I – Bad login request allows username enumeration. The log file contains passwords in clear text.

D – The user is locked out of the system after 5 bad login attempts and must be reset manually.

E – End-points for admin functions aren't secured properly.

Lack of 2FA

Category	Rating	Reason
Damage	3	Full account compromise / account take-over
Reproducibility	3	Nothing special needed to reproduce
Exploitability	1	Attacker must know someone’s username / password
Affected users	3	All users of the system are affected
Discoverability	2	Attacker would have to observe someone logging in to identify vector

3 + 3 + 1 + 3 + 2 = 12

12 = HIGH

HIGH RISK

Remediation: Enable SMS / Email / Authenticator App 2FA on the application

User Locked out after 5 Attempts

Category	Rating	Reason
Damage	2	An individual user is locked out of the system but combined with user enumeration, can deny access to multiple users
Reproducibility	3	Nothing special needed to reproduce
Exploitability	3	Very easy to exploit
Affected users	3	All users of the system are affected
Discoverability	3	Very easy to discover

2 + 3 + 3 + 3 + 3 = 14

14 = HIGH

HIGH RISK

Remediation: Block offending IP address, expire lock after 10 minutes

Summary

Insecure Design

- Defined insecure design
- Looked at the OWASP SAMM
- Examined Threat Modeling
- OWASP projects that help to mitigate

Up Next:
Security Misconfiguration