Incident Response in the SDLC

Tobias Kasch

{devday.23}

/root/00_agenda

Introduction

What is Incident Response

Responsibilities

Challenges

Preparing for Incidents in the Software Lifecycle

Honorable Mentions

Introduction

/root/01_whoami

- **Incident Responder &Forensic Analyst**
  Telekom MMS
- Former Pentester and DevSecOps Guy
- Open Source Advocate

/root/02_background

- Recent Security Breaches of IT Services Providers and Developers
  - Adesso
  - Materna
  - 3CX
  - Continental
  - WesternDigital
  - SolarWinds

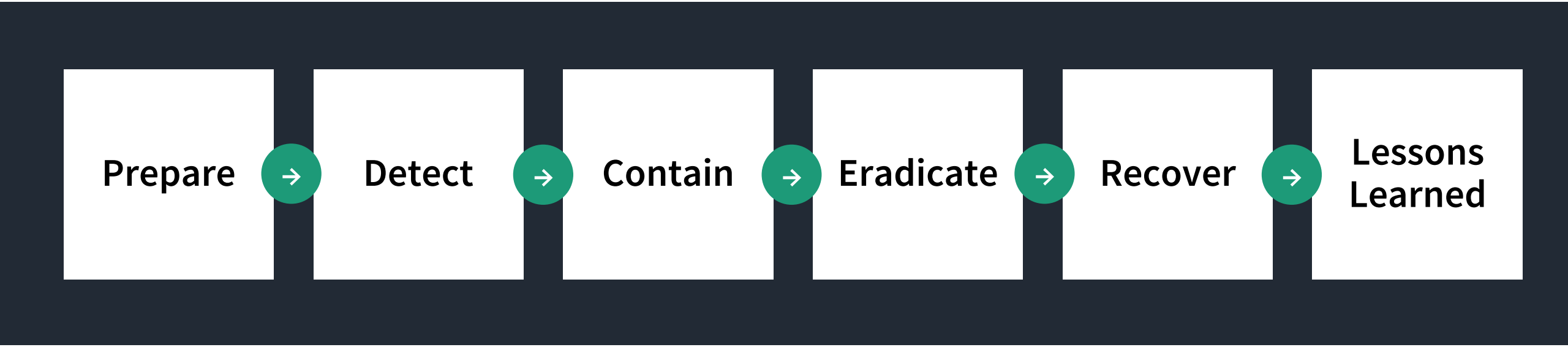
- First hand experience rebuilding complex infrastructures

/root/03_goals

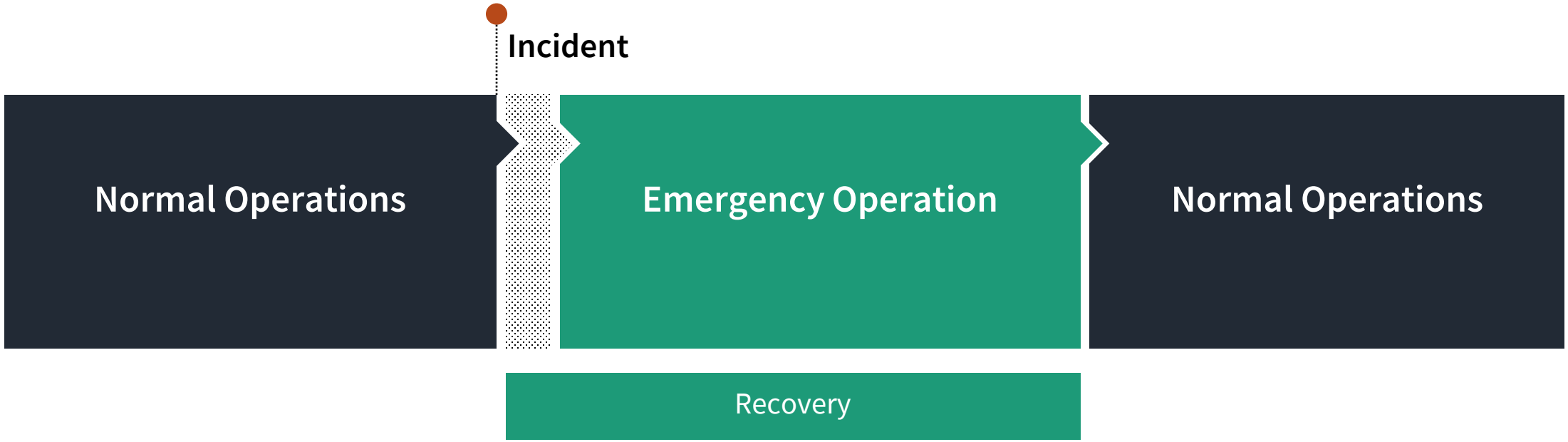
- Stressing the importance of Incident Response in designing and building software and business processes
- Understand what support and hinders Incident Response
- Starting a discussion about when and how to think about Security Breaches in the Software Community

What is Incident Response

/root/04_incident.response

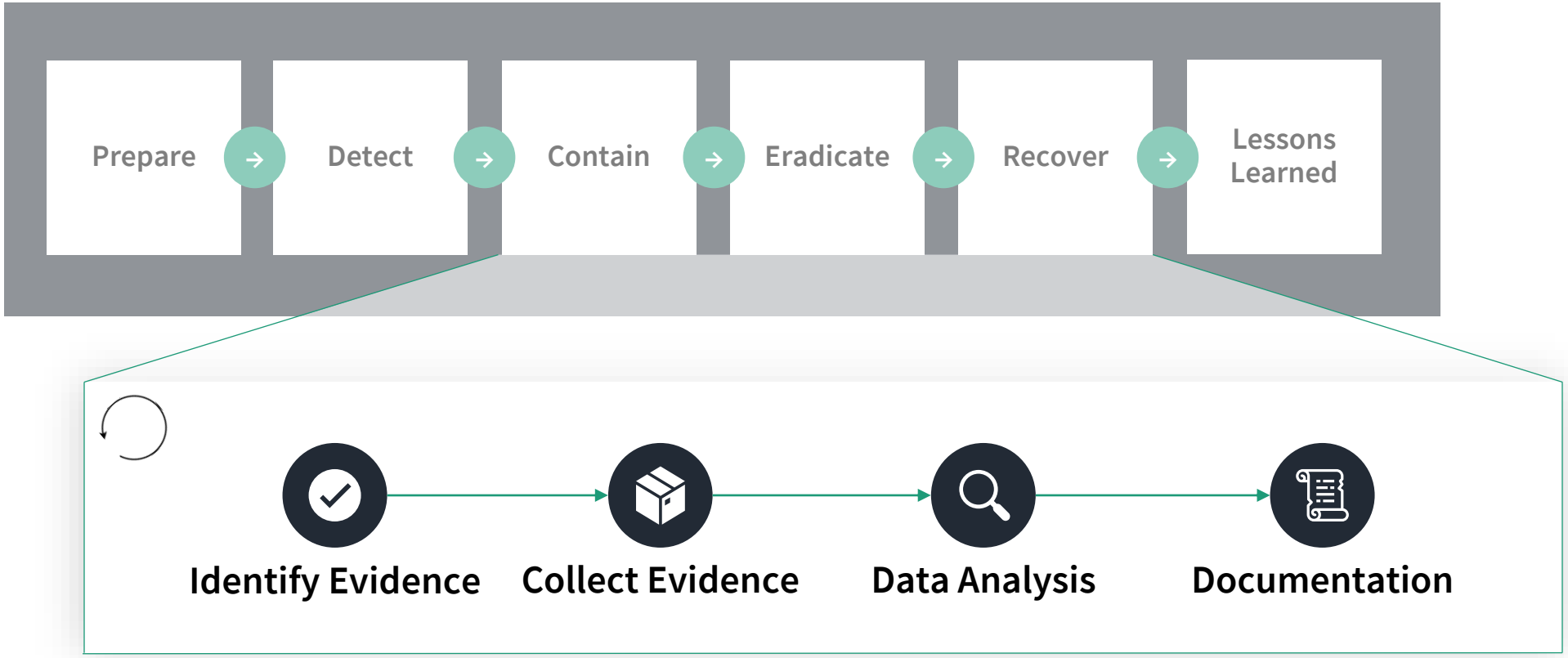


/root/05_emergency.operation



- Prioritization
- Workarounds
- Manual Tasks

/root/06_digital-forensics



/root/07_key.artefacts

- Logs, Logs and more Logs
  - Connection Logs
  - Authentication Logs
  - Audit Logs
  - Request Logs
- Filesystem information
- Memory (rare)

Responsibilities in Incident Response

/root/08_responders.responsibilities

- Responders main task is to find a way to minimize downtime, data loss and reputation loss
- They have tailored tools and processes available
- They have to work with the data that is available
- They have to work with the knowledge/documentation that is available
- They won’t have deep knowledge of each system and it’s architecture

/root/09_operations.responsibilities

- Operation teams are tasked with:
  - Supporting the investigation
  - Isolating the incident
  - Securing the infrastructure
  - Rebuilding networks and systems
- Sometimes involved in the overall response strategy
- They know the infrastructure and base systems well

/root/10_software.teams.responsibilities

- Generally not involved in the overall response process
- Sometimes “abused” for daunting manual operation tasks (installing clients)

But:

- They know the ins and outs of their applications and systems
- They know what their users expect
- They will find creative ways for challenges a responder never thought of

Challenges in Incident Response

/root/11_time

- Analysis vs. Recovery efforts have to be weighted
- Business interruptions cost real money
- Many analysis results are pre-conditions for certain remediation/recovery steps
- You will never have enough answers for all the stakeholders
- Some breaches are detected months after happening

/root/12_data.sources.and.quality

- Logs are the #1 artefact for all analysis steps yet they are often:
  - Not well documented
  - Not centrally available
  - Not secured from manipulation
  - Not configured correctly
  - Not retained long enough
  - Not known
  - Not available at all
  - Behind vendor lock (proprietary formats)

/root/13_available.documentation

- Missing documentation means Trial and Error
  - Which business processes and system are needed?
  - How can they be recovered?
  - What actions are logged?
  - Where are logs stored?
  - Who is responsible for what?
  - How can data be migrated?
  - How can data be sanitized?
  - …

/root/14_available.expertise

- You will always have to few staff at hand
- You will have to rely on external contractors
- Coordination & Communication costs time and resources

/root/15_learnings

- Prepare for breaches on each abstraction layer of the business as early as possible
- Include Blue Team members early in all decision making
- Make use of all the skills available in your teams
- Do not reinvent the wheel except it’s absolutely necessary

Preparing for Incidents in the Software Lifecycle

/root/16_state.of.the.art

- Software engineers are disengaged from the incident response (IR) structures
- Information exchange between these domains will increase software resilience and IR capabilities
- There is no blueprint framework available for incorporating these domains
- There are frameworks available to think IT Security in the software lifecycle

/root/17_microsoft.sdl

- Security Training

- Establish Security Req.
- Create Quality Gates
- Conduct Risk Assessment

- Define Security Requirements
- Threat Modelling
- Analyse Attack Surface

- Use approved tooling
- Deprecate unsafe functions
- Static Analysis

- Dynamic Analysis
- Fuzz Testing
- Attack Surface Review

- Create Incident Response Plan
- Final Security Review

- Execute Incident Response Plan

/root/18_microsoft.sdl

- Security Training
- Joined Lessons Learned

- Establish Security Req.
- Create Quality Gates
- Conduct Risk Assessment
- Create Abuse Cases

- Define Security Requirements
- Threat Modelling
- Analyse Attack Surface

- Use approved tooling
- Deprecate unsafe functions
- Static Analysis
- Create Incident Response Plan

- Dynamic Analysis
- Fuzz Testing
- Attack Surface Review
- Test Incident Response Plan

- Final Security Review
- Dedicated IR information packages

- Execute Incident Response Plan

/root/19_joined.lessons.learned

- Conduct Lessons Learned session with development teams about real world incidents
- Helps understanding risks and threats
- Implementation of auditing and alerting capabilities needs software changes
- Insights from the development teams allow for customized detection rules

/root/20_establish.security.requirements

- Think whether incorporating an business continuity scenario is necessary
  - Is my application or data critical to the business
  - Is access to the data a critical threat
  - How long could the application be offline without affecting the business
  - Are there manual work-arounds available
  - Etc.
- Not all applications need to be recovered in a secure manner

/root/21_abuse.cases (misuse cases)

- Describe how an attacker would misuse weaknesses in software features
- Way less generic then common security requirements

> “As an attacker, I manipulate the primary key and change it to access another's users record, allowing viewing or editing someone else's account.”

- Require knowledge in the attackers domain

/root/22_abuse.cases (misuse cases)

- Not every abuse case may be preventable.
  E.g. Using stolen valid credentials.
- For these cases think about how detection, reaction and remediation to this scenarios may look like and what feature are needed to enable these steps
- For example:
  - Add authentication attempts and user actions to the audit log
  - Include IP / User agent / Client ID and other data to allow mapping of actions
  - Allow for central password reset and account deactivation via the admin menu

/root/23_define.security.requirements

- Include wishes from Responders to the requirements list:
  - Create logs for CRUD actions
  - Export of log files in standardized machine readable format
  - Data import, export and sanitization
  - Documentation of data structures and where data is ingested, processed and presented back to the user
  - Define unsafe system states (which files are never suspect to change, which configuration should not be changeable, etc.)
  - Plan for a minimal viable emergency operation
  - Support data collection and analysis tools of your responders

  …

/root/24_IR.plan

- Create an Incident Response Plan for your application
  - What minimal requirements must be met for the application to run
  - Where are logs and configs found, how can they be exported
  - How are these structured
  - Where is data stored and in which format
  - What are expected and unexpected system states and data points
  - How can data/setting be sanitized and imported
  - How can user be exported/imported
  - How can keys be rotate
- Make a dry run recovery test for with your plan and your operations

/root/25_IR.publish

- Make all necessary documentation available to the responsible teams
- Make sure to keep all information up to date and allow for offline storage (USB Drives etc.)
- Make clear what is documented and where it can be found

Some honorable mentions

- Atlassian Disaster Recovery Guides
- Atlassian Bitbucket Auditing
- MSSQL Business Continuity Guide
- Azure/AWS Response Guide

/root/26_wrap.up

- Large-Scale Security Incidents require a lot of preparation
- Resources and Workforce are always short in IR scenarios
- Thinking IR while designing software drastically shortens outages and enables analysts
- Not every Software requires continuity management
  - Propper business risk analysis is key

Ask away!

https://github.com/Explie/presentations/

/root/99_references

https://resources.sei.cmu.edu/asset_files/WhitePaper/2013_019_001_299145.pdf

https://cheatsheetseries.owasp.org/cheatsheets/Abuse_Case_Cheat_Sheet.html#step-1-preparation-of-the-workshop

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1514408

https://www.microsoft.com/en-us/securityengineering/sdl/practices

https://owasp.org/www-pdf-archive/SDL_in_practice.pdf

https://learn.microsoft.com/en-us/sql/database-engine/sql-server-business-continuity-dr?view=sql-server-ver16

https://confluence.atlassian.com/enterprise/disaster-recovery-for-atlassian-data-center-892801335.html

https://learn.microsoft.com/en-us/security/operations/incident-response-overview

https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/introduction.html