6 Release

This checklist is dynamic, not exhaustive, and will be updated regularly. If you have any suggestions or comments, we would like to hear from you

How to create a software-related incident response plan?

- Create an incident response plan of the software to be released. This should include impact assessment, measures, and continuous improvement of the software.
- Establish a contact point or response centre with its own communication channels for reporting incidents, taking into account internal and external incident reporting. For example, encouraging and having a good dialogue with "whistle-blowers" will be crucial to whether or not users will report vulnerabilities, deviations, and errors. User reports of security incidents can help increase the robustness of the software if these incidents are properly managed.
- To deal with future threats to the software, the plan must cover:
    - Contact information and communication channels
        - in the case of data protection and/or security breaches
        - for technical support
- Binding agreements that specify suitable response times for relevant suppliers
- Establish risk management procedures for the various scenarios described in the requirements phase for risk assessment of data protection and security. Who easy and likely is it that different scenarios occur, what is the impact, who should be informed, should the system be turned off immediately, what is mandatory logging and triggering of alarms, etc. Example:
    - What is the consequence if you are too proactive and shut down a system when an incident occurs?
    - Is it likely that evidence concerning alarm triggers (thresholds) is removed upon multiple login attempts by the same user from multiple IP addresses, many users from the same IP address, etc.?
    - What should happen if someone discovers sensitive personal data being stored in the wrong place? Who should they contact?
- Definition of what the plan covers, and what qualifies as an incident.
- Definition of the life cycle of a deviation, as well as procedures for detecting, analysing, reporting, handling, and normalising.
    - Detect
        - Security monitoring of servers, end points, and network should discover suspicious activity that could exploit vulnerabilities in the software, resulting in data protection and security incidents.
        - Security monitoring and alarms triggered by suspicious patterns help to provide an overview of threats, vulnerabilities, and security incidents.
    - Analyse and Verify
        - Analyse the incident, review the logs, gain an overview of what happened, and the scope of the incident.
        - Verify if suspicious activity is an actual security breach or a false positive.

        - Follow the company procedure for forensics, including who will lead the investigation, when the police or other experts will be called upon for assistance, etc.
    - Report
        - Report security breaches according to internal guidelines. Note that warning of a deviation can also alert the attackers, so be careful how you raise the alarm.
        - Report deviations that include personal data breaches of confidentiality, integrity, and availability to the Data Protection Authority *within* 72 hours.
        - Inform the data subject of deviations that include breaches of confidentiality, integrity, and access to personal data.
        - Consider sharing the resulting experiences with the industry or sector
        - Should other authorities be informed, such as the police, The Norwegian National Security Authority (NSM), The Norwegian Centre for Information Security (NorSIS), Financial Supervisory Authority of Norway (Finanstilsynet), Norwegian Board of Health Supervision (Helsetilsynet) etc.
    - Handle
        - Security incidents should be handled according to the business continuity plan. Are deviations handled differently depending on whether they have a direct impact on the customer or if the effects will be internal within the company? It should be noted that an unplanned patch can serve as a warning to potential attackers that a vulnerability exists.
        - Implement vulnerability reducing measures, such as patching, changing procedures, extended logging, regulate access, closing ports, etc.
        - Test if the implemented measures are working as intended, and are not causing new vulnerabilities.
    - Normalise
        - Restore management, operation, and maintenance to their normal state.
- Description of the configuration and handling of logs, including guidelines from the data protection regulation.
- Perform an evaluation of the incident response plan, and how the experiences can have consequences for the development process, the company, and others affected.
- Stakeholders must be identified, and a procedure should be established to describe when and how they will be informed. Notification of personal data breaches to the Data Protection Authority (within 72 hours) and communication to the data subjects.
- Recommended procedures for patching the developed software, including related software (including those from third parties). The procedures must form part of the company's overall plan.

The business continuity plan should be updated according to the requirements above. Events that could trigger updates include new services, altered criticality, changes to the emergency shutdown procedure, changes to the criticality matrix, changes to alert lists, etc.

Full security review of developed software

- The security review should be based on reviews carried out during the previous phases of the development process.

- Use different expert groups for the review, discuss different scenarios, and assess the consequences and possible measures.
- Consult with roles/persons who are directly involved in the activities (what has worked well, and what has not worked well – continuous improvement).
- A full security review should be included in the control gates and be carried out prior to release. All activity should be reviewed to reveal any deviations, and should include
    - data protection requirements
    - security requirements
    - acceptable level of risk
    - the results of the data protection impact assessment (DPIA)
    - risk assessment, including risk treatment plan and risk acceptance
    - attack surfaces
    - output from tools
    - design analysis (Are there any deviations from the planned and currently applicable design specifications?)
    - code analysis (Are manual and static analyses performed in accordance with the revealed focus areas, are the results aligned with tolerance levels?)
    - tools and third-party components (Are only authorised tools and third-party components used during the development process?)
    - dynamic testing, penetration testing or vulnerability analysis (Are the findings aligned with the defined tolerance levels?)
    - the results of penetration tests and vulnerability analysis
    - a review of the finished software against the original threat models, to uncover implementation anomalies and, if necessary, put into place measures to handle them

Approval of release and archiving

- The software must be approved before release, to ensure that data protection and security requirements are met.
- Approval should be carried out by a manager with the responsibility/authorisation to do so, with support from stakeholders in relevant disciplines
- All relevant data from the entire development process should be archived, including all specifications, source code, binary code, private symbols, data protection impact assessment (DPIA), risk assessments, documentation, business continuity plans, licenses, and terms of service for third-party software. Archiving is important for future needs of support tasks, reduce long-term software management and maintenance costs, and enabling rollback to a previous version. For example, archiving can be carried out in Escrow.
- Based on the results of a full security review, the responsible/authorised manager will decide if the software should be released. (Persons who are not responsible can provide recommendations for release).

Why set requirements for release management?

- There are requirements for incident management and business continuity. Having a plan for incident management may help to handle serious incidents properly and

efficiently, and to protect against adverse consequences from errors or accidents, cf. Articles 32, 33, and 34.

- The final security review of the software must be committed in the internal control system or the information security management system (ISMS), cf. articles 30 and 32.
- The release must be approved, and archiving of the activities carried out during development, cf. articles 30 and 32.