Cyber Security Guideline for Software & System Development at Blickfeld GmbH

Document revision 2.0, 19.02.2025

Revision history

Rev.	Date	Authors	Reviewed by	Approved by
1.0	14.06.2023	J. Teiwes	R. Wojtech	M. Müller
1.1	01.07.2024 #54276	J. Teiwes	R. Wojtech	M. Müller
2.0	19.02.2025 #55179	J. Teiwes	R. Wojtech	M. Müller

1 Introduction

The purpose of this Cyber Security Policy for Software Development is to ensure that all software developed or maintained by Blickfeld GmbH adheres to best practices in cybersecurity. This policy aims to protect the integrity, confidentiality, availability and safety of software and hardware products offered by Blickfeld GmbH, as well as the data it processes.

2 Scope

This policy applies to all software development activities conducted by Blickfeld GmbH, including general software development, embedded software development, and the use of third-party software components.

3 Objectives

- Safeguard the software development lifecycle (SDLC) from cybersecurity threats.
- Ensure compliance with relevant laws, regulations, and standards (e.g. DSGVO).
- Protect Blickfeld GmbH's data and resources from unauthorized access and breaches.
- Promote a culture of security awareness among developers and stakeholders.

4 Roles and Responsibilities

4.1 Cybersecurity Team

- Develop and maintain security policies, standards, and guidelines.
- Conduct regular security assessments and audits.

4.2 Development Team

- Participate in internal and external security training and awareness programs.
- Document and raise bugs or vulnerabilities found during testing or fuzzing.
- Collaborate with the Cybersecurity team during security assessments.
- Provide training and resources for secure coding practices.
- Strict four-eyes code review policy for all changes made to new or existing software components.
- Actively follow CVE (Common Vulnerabilities and Exposures) data feeds to stay informed about new vulnerabilities (e.g. https://github.com/CVEProject/cvelistV5).
- Monthly automated assessment of CVEs which might affect software components shipped with Blickfeld Lidar sensors.

4.3 Testing Team

- Ensure that the functionality fulfills specifications on all levels from unit to system.
- Identify and report regressions and potential threats.

4.4 Project Managers

- Ensure that security requirements are documented and integrated into project plans.
- Monitor compliance with the security policy throughout the project lifecycle.

5 Secure Software Development Lifecycle (SDLC)

5.1 Planning

Security Requirements Analysis: Identify application security requirements based on data sensitivity, regulatory requirements, and threat models.

Risk Assessment: Conduct a risk assessment to identify potential security threats and vulnerabilities.

5.2 Design

Secure Architecture: Design system and software architecture with security engineering principles such as least privilege, defense in depth, and secure defaults.

Threat Modeling: Perform threat modeling to identify potential threats and design countermeasures.

5.3 Implementation

Restricted Access to Source Code: Strictly control the repositories, granting access only to authorized persons based on their roles and only when needed.

Secure Coding Practices: Adhere to secure coding guidelines to prevent common vulnerabilities such as privilege escalation, cross-site scripting (XSS), and buffer overflows.

Dedicated Development Environment: Isolate Development, testing, and production environments from each other to prevent unauthorized access, reduce security and safety risks, and ensure system integrity. Implement access controls, monitoring, and security measures to protect sensitive data.

Code Reviews: Conduct regular code reviews with a focus on security vulnerabilities.

5.4 Testing

Unit & Module Testing: Ensure that the functionality of the implementation behaves according to the specifications and runs with high performance.

Dynamic Analysis: Use the qemu tool to perform dynamic analysis to test the application in a sandboxed runtime environment for functionality and security vulnerabilities.

Fuzz Testing: Use fuzzing techniques to identify input handling vulnerabilities and parameter space boundaries for relevant software and hardware components.

Security Testing: Specifically test the software modules which implement security measures by validating their resilience against common attack vectors.

Acceptance Test Information: Document detailed results from acceptance tests and communicate them to the relevant stakeholders to ensure continuous secure operation and performance in existing deployments.

5.5 Release

Release process: The release process ensures high quality and highly functional firmware versions for all Blickfeld GmbH Lidar sensors. All internal stakeholders test & approve before a new version is created.

Publishing: New firmware versions are published on the Blickfeld website and github repository.

5.6 Maintenance

Monitoring and Logging: Implement monitoring and logging to detect and respond to security incidents.

Vulnerability Management: Continuously monitor for new vulnerabilities and apply necessary patches or mitigations.

CVE Tracking: Developers actively follow CVE mailing lists and ensure that vulnerabilities are addressed promptly. Conduct monthly automated builds to document unpatched and patched CVEs, ensuring transparency and accountability in handling vulnerabilities.

Patch Management: Implement a process for timely application of security patches and updates.

6 Third-Party Components

Security Assessment: Conduct a security assessment of third-party components before integration.

Vendor Management: Ensure that third-party vendors comply with Blickfeld GmbH's security requirements.

Licensing Compliance: Verify that the use of third-party components complies with licensing terms and conditions and don't use viral copy-left open source licenses.

Closed Source Components: Do not use any closed source or precompiled / binary software components.

7 Incident Response

Incident Detection: Implement mechanisms for detecting security incidents in software applications. Indications for possible security breaches come from the constant monitoring and logging facilities as well as from customer claims or support requests.

Incident Management: Establish an incident response plan for handling security incidents, including identification, containment, eradication, recovery, and lessons learned.

Communication Plan: Develop a communication plan for notifying stakeholders, including users, management, and regulatory bodies, in case of a security breach.

8 Training and Awareness

Developer Training: Provide regular training on secure coding practices, emerging threats, and new security tools.

Security Awareness: Promote security awareness among all employees involved in software development through workshops, seminars, and regular updates.

Locked Desks: All personnel of Blickfeld GmbH are encouraged to lock their workstations or laptops so that nobody can access their account or take any actions on their behalf.

Social engineering Awareness: The Cybersecurity team informs the staff about active social engineering campaigns and educates the teams about threat vectors originating from daily tasks (e.g. opening email attachments).

Restricted Access: Only members of the IT team are allowed to enter the server room of Blickfeld GmbH.

9 Policy Review and Updates

Yearly Review: Review and update this policy periodically to address emerging threats and changes in regulatory requirements.

Stakeholder Feedback: Incorporate feedback from developers, Cybersecurity teams, and other stakeholders in policy updates.