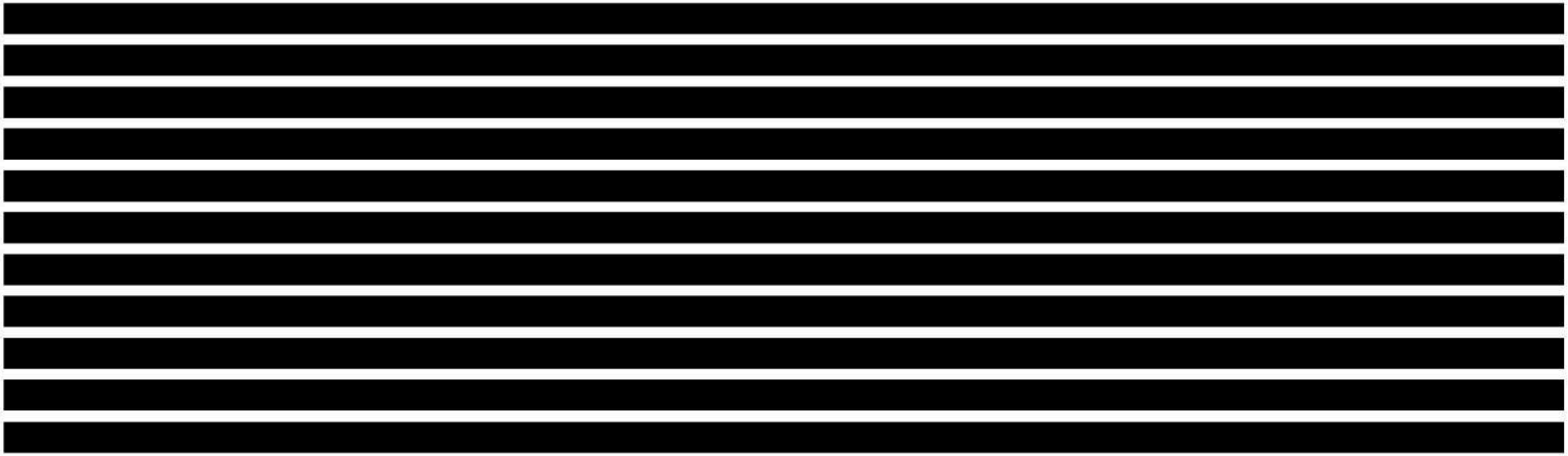
Loda Finance
PUBLIC VERSION

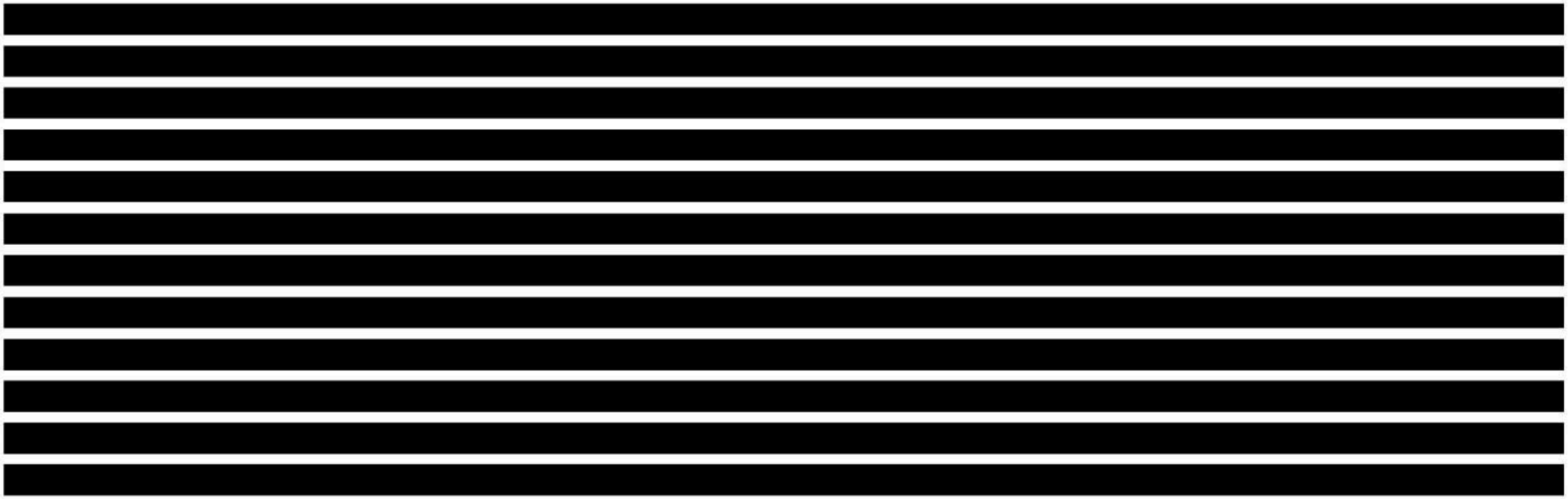
ARCHITECTURE & CODEBASE SECURITY AUDIT

05.12.2021

Made in Germany by Chainsulting.de

Table of contents





1. Disclaimer

The audit makes no statements or warrantees about utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only.

The information presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of Loda Finance Pty Ltd. If you are not the intended receptor of this document, remember that any disclosure, copying or dissemination of it is forbidden.

Major Versions / Date	Description
0.1 (22.11.2021)	Layout
0.2 (24.11.2021)	Test Deployment
0.5 (26.11.2021)	Automated Security Testing Manual Security Testing
0.6 (28.11.2021)	Development Productivity (Code Conventions Check, packages)
0.7 (01.12.2021)	Performance (Connection pooling, caching, Transaction Concurrency)
0.8 (02.12.2021)	Maintainability & Ease of Deployment
0.9 (03.12.2021)	Summary and Recommendation
1.0 (03.12.2021)	Final document

2. About the Project and Company

Company address:

Loda Finance Pty Ltd
ACN 648 427 721
Unit 506, 113 Commercial Road
Teneriffe, QLD, 4005
Australia

Website: https://loda.finance

Twitter: https://twitter.com/loda_fi

2.1 Project Overview

Loda is the only Australian platform that allows you to collateralize your crypto and borrow AUD instantly. Lodas interest rates are competitive, if not the lowest anywhere across the space. Using crypto as collateral is also the most tax efficient way to access fiat, as you are not selling the underlying asset, and therefore not incurring a capital gains event.

Loda wants to re-imagine consumer lending using crypto as collateral. They believe digital assets have an enormous potential to re-shape all financial products, with the guiding principle to make markets fairer and more transparent.

The safety of clients funds is the main priority for Loda. All users' assets are stored in bank-grade Class III vaults while the partners' hardware and software guarantee allows for instant access to the funds and full independence from third-party providers. The system are ISO 27001:2013 Compliant and they maintain regular AML & KYC Checks.

3. Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 – 10	A vulnerability that can disrupt the code functioning in a number of scenarios or creates a risk that the code may be broken.	Immediate action to reduce risk level.
High	7 – 8.9	A vulnerability that affects the desired outcome when using a codebase, or provides the opportunity to use an application in an unintended way.	Implementation of corrective actions as soon as possible.
Medium	4 – 6.9	A vulnerability that could affect the desired outcome of executing the code in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the code and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

4. Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pen testers and developers, documenting any issues as there were discovered.

4.1 Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
   i. Review of the specifications, sources, and instructions provided to Chainsulting to make sure we understand the size, scope, and functionality of the codebase.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Chainsulting describe.

2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   ii. Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.

3. Best practices review, which is a review of the codebase to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

4. Specific, itemized, actionable recommendations to help you take steps to secure your codebase.

4.2 Codebase Overview

Source: [redacted]

Client

Total : 135 files, 9570 codes, 56 comments, 1255 blanks, all 10881 lines

Languages

language	files	code	comment	blank	total
[redacted]					
[redacted]					
[redacted]					
[redacted]					
[redacted]					
[redacted]					
[redacted]					
[redacted]					
[redacted]					

Core-Backend

Total : 4 files, 140 codes, 1 comments, 20 blanks, all 161 lines

Languages

language	files	code	comment	blank	total
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Data

Total : 32 files, 1513 codes, 75 comments, 164 blanks, all 1752 lines

Languages

language	files	code	comment	blank	total
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Server

Total : 92 files, 5889 codes, 84 comments, 920 blanks, all 6893 lines

Languages

language	files	code	comment	blank	total
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Server-interfaces

Total : 7 files, 308 codes, 0 comments, 54 blanks, all 362 lines

Languages

language	files	code	comment	blank	total
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

5. Scope of Work

The Loda Finance Team provided us with the files that needs to be tested. The scope of the audit is the Loda Finance infrastructure.

1. Automated Vulnerability Test (OWASP, Acunetix, Sonarsource, etc.)
2. Manual Security Testing (Line by line, Overflow, CVE, etc.)
3. Test environment deployment
4. Evaluating and testing software architecture
   - Development Productivity (Code Conventions Check, packages)
   - Functions & Logic Testing
   - Performance (Connection pooling, caching, Transaction Concurrency)
   - Reliability & Availability
   - Maintainability & Ease of Deployment

The main goal of this audit was to make sure the infrastructure is built according to newest standards and securely developed. The auditors can provide additional feedback on the code upon the client's request.

5.1 Manual and Automated Vulnerability Test

Details
Web Target: [redacted]
Source: [redacted]

CRITICAL ISSUES

During the audit, Chainsulting's experts found **no Critical issues** in the code of the smart contract.

HIGH ISSUES

5.1.1 [redacted]
Severity: HIGH
Status: [redacted]
Code: [redacted]
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

MEDIUM ISSUES

5.1.2 [redacted]

Severity: MEDIUM

Status: [redacted]

Code: [redacted]

Target: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.3 [redacted]
Severity: MEDIUM
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

[redacted]		

5.1.4 [redacted]
Severity: MEDIUM
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.5 [redacted]
Severity: MEDIUM
Status: [redacted]

Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.6 [redacted]
Severity: MEDIUM
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

LOW ISSUES

5.1.7 [redacted]
Severity: LOW
Status: [redacted]
Code: [redacted]
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.8 [redacted]
Severity: LOW
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.9 [redacted]
Severity: LOW
Status: [redacted]
Code: [redacted]
Target: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

		[redacted]

5.1.10 [redacted]
Severity: LOW
Status: [redacted]
Code: [redacted]
Target: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.11 [redacted]
Severity: LOW
Status: [redacted]
Code: [redacted]
Target: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.12 [redacted]
Severity: LOW
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

[redacted]		

5.1.13 [redacted]
Severity: LOW
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

[redacted]		

5.1.14 [redacted]
Severity: LOW
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

INFORMATIONAL ISSUES

5.1.15 [redacted]
Severity: INFORMATIONAL
Status: [redacted]

Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.16 [redacted]
Severity: INFORMATIONAL
Status: [redacted]
Code: [redacted]
Target: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.17 [redacted]
Severity: INFORMATIONAL
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.18 [redacted]
Severity: INFORMATIONAL
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.19 [redacted]
Severity: INFORMATIONAL
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

[redacted]		

5.1.20 [redacted]
Severity: INFORMATIONAL
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.21 [redacted]
Severity: INFORMATIONAL
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

[redacted]		

5.1.22 [redacted]
Severity: INFORMATIONAL
Status: [redacted]
Code: NA
File(s) affected: [redacted]

Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.1.23 [redacted]
Severity: INFORMATIONAL
Status: [redacted]
Code: NA
File(s) affected: [redacted]

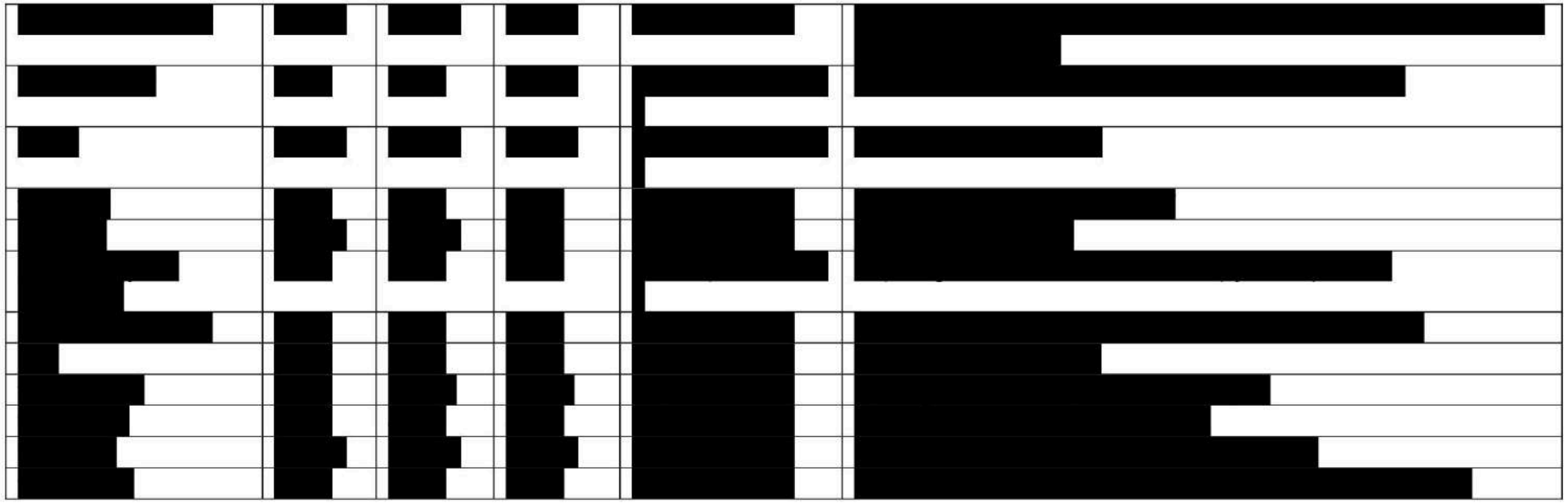
Attack / Description	Code Snippet	Result/Recommendation
[redacted]	[redacted]	[redacted]

5.2 Packages

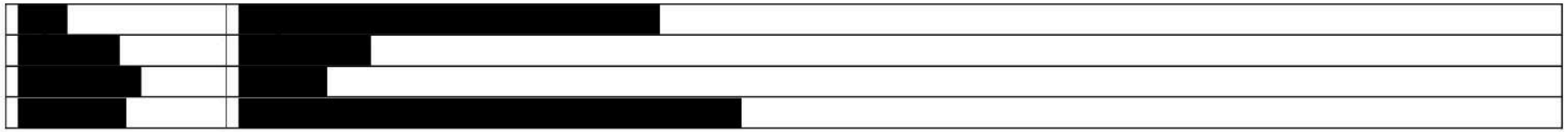
Client

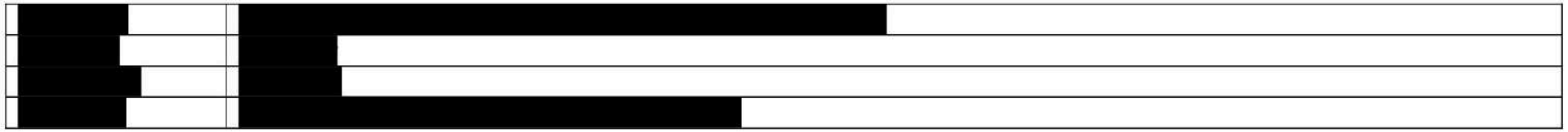
Outdated packages

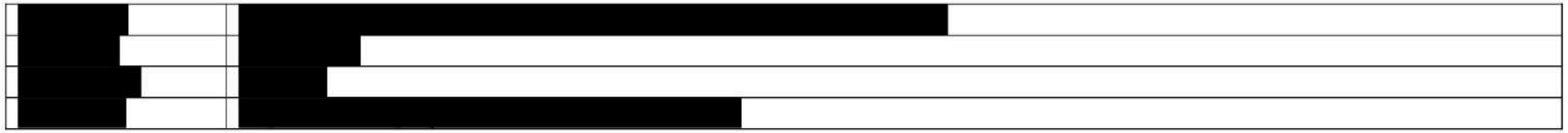
Package	Current	Wanted	Latest	Package Type	URL
[redacted]					

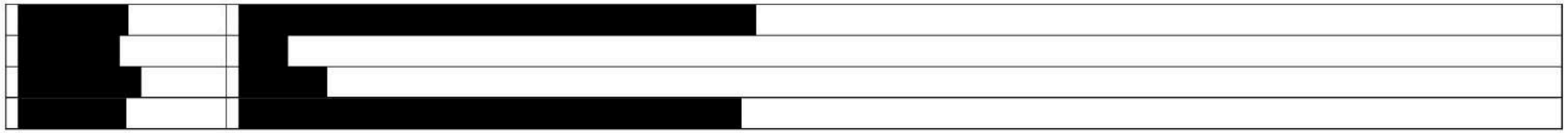


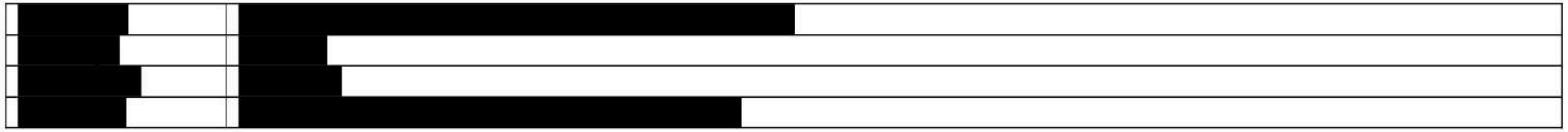
Found vulnerabilities

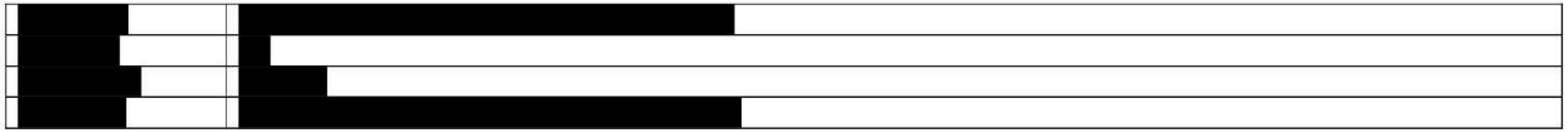


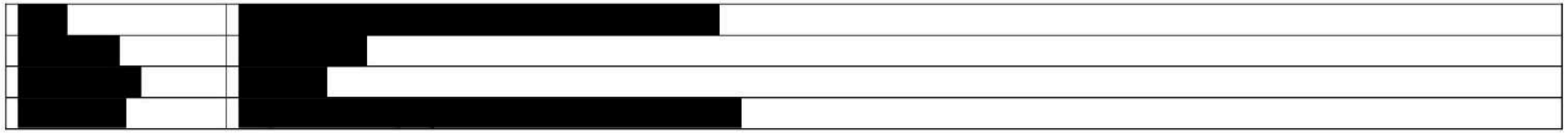


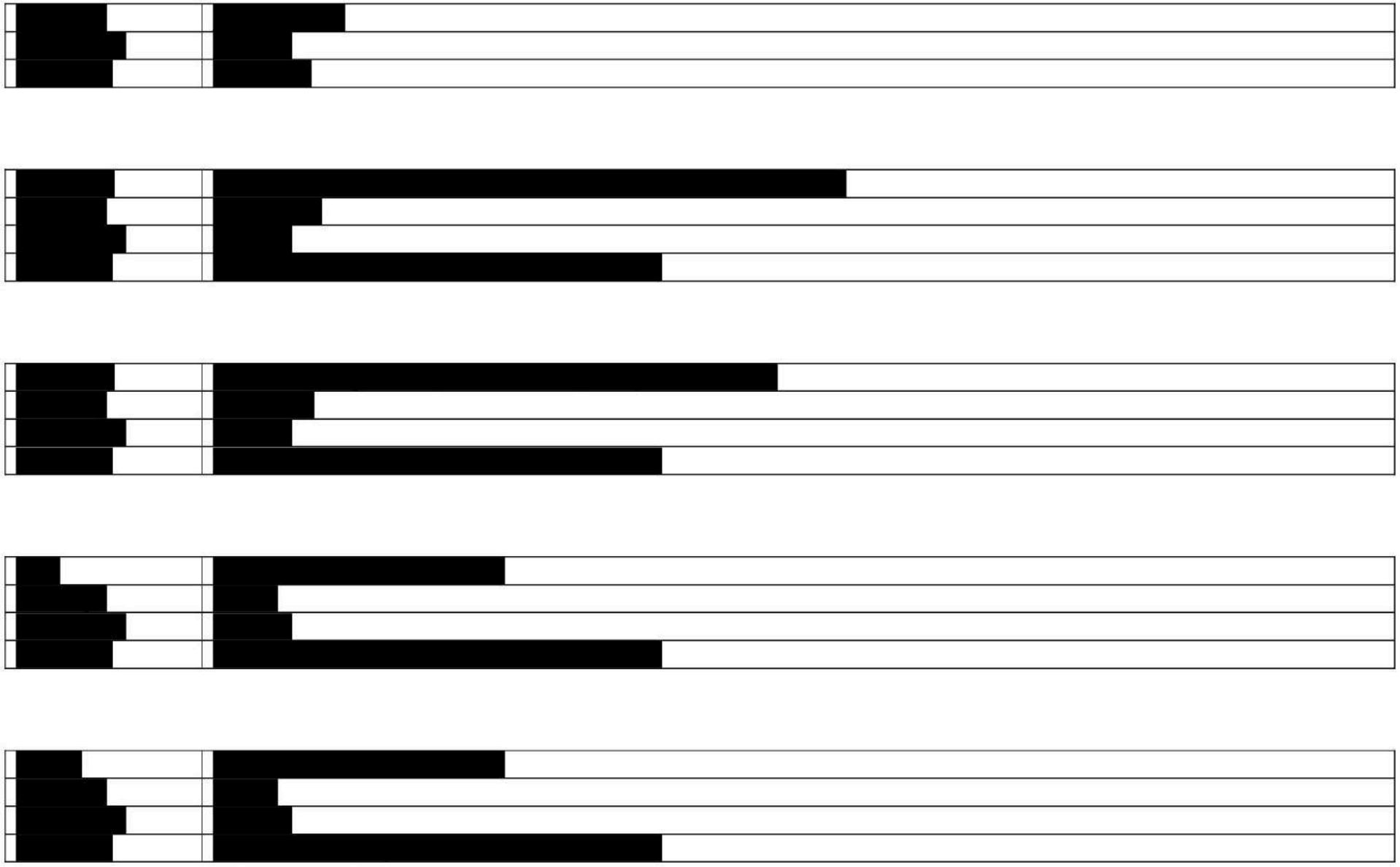


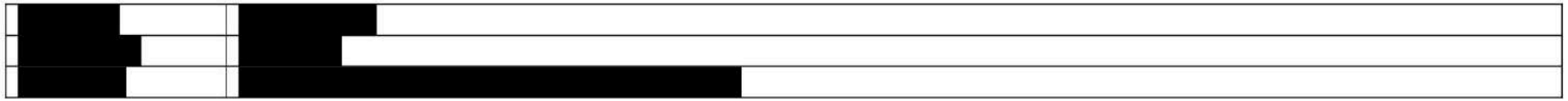


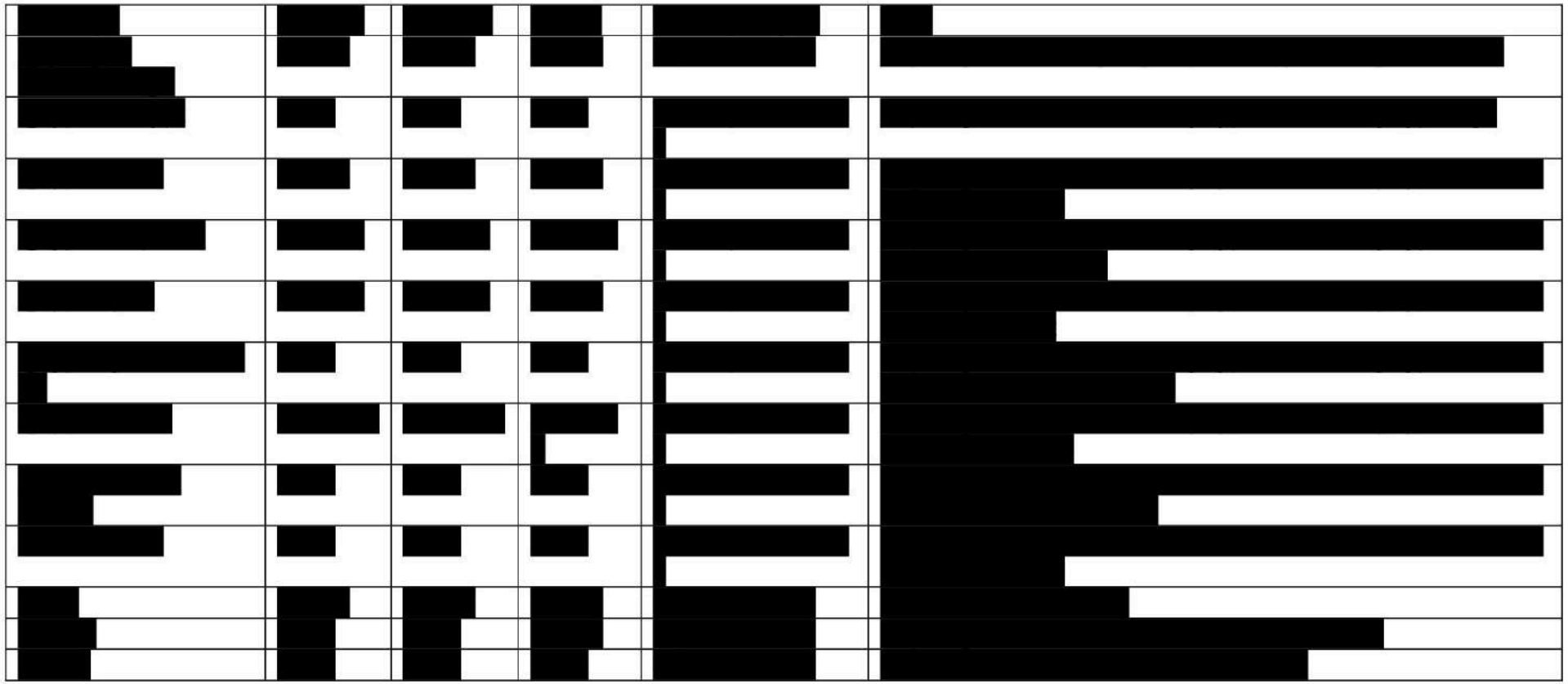


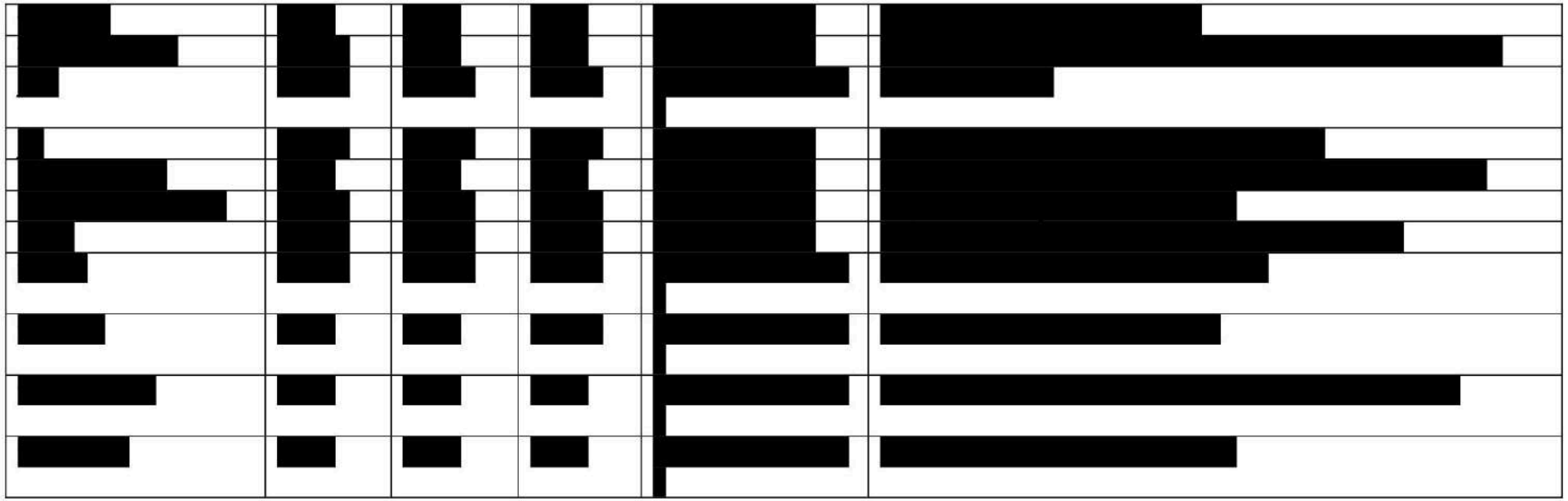




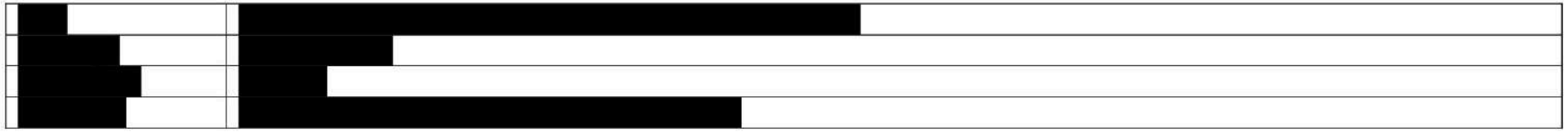


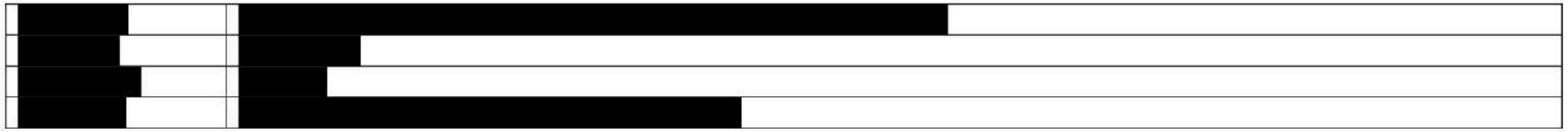


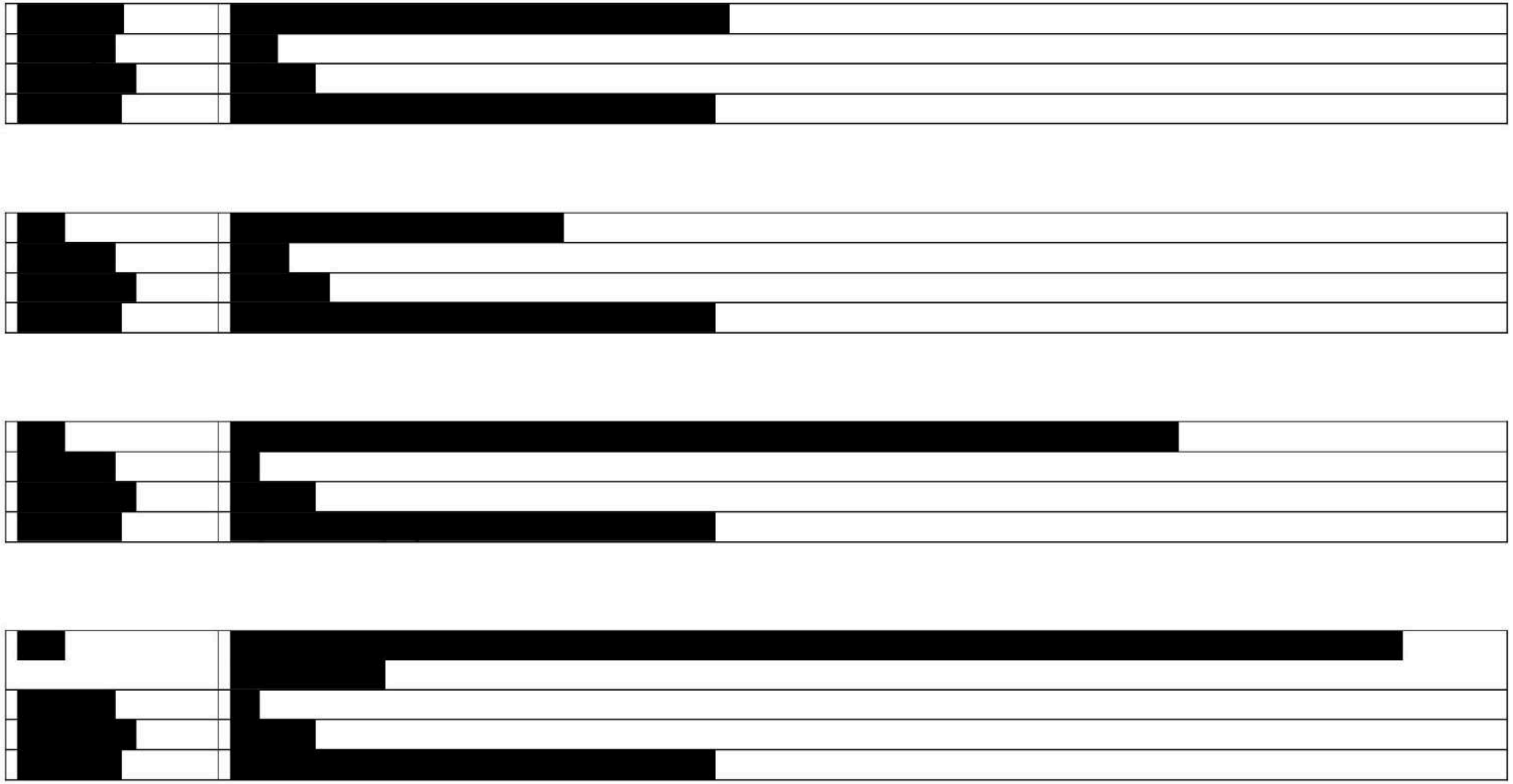




Found vulnerabilities







5.3 Performance Check

GTmetrix Grade ?

A	Performance ?	Structure ?
	98%	99%

Web Vitals ?

LCP ?	TBT ?	CLS ?
996ms	0ms	0

Performance Metrics

The following metrics are generated using Lighthouse Performance data.

First Contentful Paint ?	Good - Nothing to do here 900ms	Time to Interactive ?	Good - Nothing to do here 900ms
Speed Index ?	Good - Nothing to do here 916ms	Total Blocking Time ?	Good - Nothing to do here 0ms
Largest Contentful Paint ?	Good - Nothing to do here 996ms	Cumulative Layout Shift ?	Good - Nothing to do here 0

GRADE		SUGGESTION
F	45	Add Expires headers

Web pages are becoming increasingly complex with more scripts, style sheets, images, and Flash on them. A first-time visit to a page may require several HTTP requests to load all the components. By using Expires headers these components become cacheable, which avoids unnecessary HTTP requests on subsequent page views. Expires headers are most often associated with images, but they can and should be used on all page components including scripts, style sheets, and Flash.

A	96	Make fewer HTTP requests
A	100	Avoid empty src or href
A	100	Put JavaScript at bottom
A	100	Reduce the number of DOM elements
A	100	Make favicon small and cacheable
A	100	Avoid HTTP 404 (Not Found) error

6. Executive Summary

Two (2) independent Chainsulting experts performed an unbiased and isolated audit of the architecture and codebase. The final debriefs took place on the December 02, 2021.

The main goal of this audit was to make sure the infrastructure is built according to newest standards and securely developed. During the audit, no critical issues were found, after the manual and automated security testing. Overall, the code quality and architecture had a high grad of professionalism.

We recommend the following things to apply to the next dev ops:

1. Check our issues and get them fixed
2. Make sure newest packages are used, wherever possible
3. Use linter for static code analysis to flag programming errors, bugs, stylistic errors and suspicious constructs
4. Use a load balancer and DDoS protection such as Cloudflare
5. Make sure the whole architecture is documented (Software Stack, Used development tools, Architecture schemas, Database schmemas, Design Guidelines, How to guides etc.) as the readme was pretty thin for such a big codebase.
6. Make sure major codebase files has inline documentation