CONFIDENTIAL AND PROPRIETARY

authentik Cobalt test instance Penetration Test Report - September 2024

Report generated on Nov 28 2024

Prepared for Authentik Security by Cobalt. This report is for informational purposes, not for any other purpose, and may only be shared with third-parties by Authentik Security. Cobalt disclaims all liability to any third-party arising from this report. Usage of this report by shall be subject to Cobalt's terms, available at https://cobalt.io/terms/.

Targets

https://cobalt.pr.test.goauthentik.io/

https://cobalt.pr.test.goauthentik.io/api/v3/

Test period	Status
Sep 2, 2024 Sep 16, 2024	Final

Test performed by

Avanish Pathak **Lead**

Mesut Türk **Pentester**

Samandeep Singh **Pentester**

Contents

Executive Summary

Cobalt conducted a pentest of the authentik Cobalt test instance application and API to assess the risk posture and identify security issues that could negatively affect Authentik Security's data, systems, or reputation. The scope of the assessment covered authentik Cobalt test instance and included credentials for various levels of privilege within the scope. A Cobalt pentest team of 3 conducted this engagement between Sep 2, 2024 and Sep 16, 2024.

This pentest was a manual assessment of the security of the application's functionality, business logic, and vulnerabilities, such as those cataloged in the Open Web Application Security Project (OWASP) Top 10. The assessment also included a review of security controls and requirements listed in the OWASP Application Security Verification Standard (ASVS). The pentesters leveraged tools to facilitate their work. However, the majority of the assessment involved manual analysis.

During this engagement, Cobalt's testers reported their findings and rated them based on the following severities:

	●CRITICAL	●HIGH	●MEDIUM	●LOW	●INFO
Open	0	0	0	0	0
Resolved	0	0	0	5	1
Total	0	0	0	5	1

During testing, Cobalt's pentesters tested for vulnerabilities and rated them based on the following categories:

Cobalt's pentesters conducted a comprehensive security assessment of the Authentik application, a self-hosted identity provider utilizing the open-source project Authentik. The primary objective of this test was to ensure that the application adheres to industry security standards and to identify any potential vulnerabilities that attackers could exploit. During the assessment, no high or critical-risk vulnerabilities were discovered. The pentesters found that the Authentik Security team implemented robust and up-to-date security practices throughout the application.

Cobalt’s pentesters examined the internet-facing Authentik Security's self-hosted instance to identify and exploit potential security weaknesses that could allow an attacker to gain unauthorized access or perform malicious activities. These efforts aimed to simulate real-world attack scenarios, seeking vulnerabilities that could compromise the app’s confidentiality, integrity, or availability. Testers found that the application had several security best practices in place and demonstrated resilience against multiple vulnerabilities.

During the assessment, it was discovered that the application was vulnerable to HTML injection through user-supplied names in the Flow section. This issue arises because the application did not properly sanitize or escape HTML input when displaying user-entered names. As a result, an attacker could inject arbitrary HTML or JavaScript code into the application, potentially leading to manipulation of the web page or execution of malicious scripts in the context of the user's session.

The pentesters discovered that the application was susceptible to insecure file upload and stored Cross-Site Scripting (XSS) vulnerabilities by uploading crafted SVG files. The application allowed users to upload files, including SVG images, which are then used as application icons. Since the application did not properly validate or sanitize these files, an attacker could upload an SVG file containing embedded malicious scripts. This could lead to stored XSS, where the malicious script is executed in the context of other users viewing the icon.

Cobalt pentesters discovered that the application was vulnerable to stored XSS through footer links. The footer section of the application accepted and displayed user-provided links without proper sanitization. This could allow an attacker to inject malicious scripts into these links, which are then stored and executed when other users access the footer links, leading to potential script execution in the context of the victim’s session.

It was also discovered that the application enforced a weak password policy. The policy allowed users to create passwords that lack complexity and are easily guessable. This deficiency reduces the overall security of user accounts by making them more susceptible to brute-force and dictionary attacks, as users can create passwords that do not meet standard security requirements.

The pentesters discovered that the application lacked a Content Security Policy (CSP) header. The absence of a CSP header means that the application lacks a mechanism to restrict sources of content and scripts, which can expose it to XSS attacks and other forms of content injection. Implementing a CSP would help mitigate the risk of such vulnerabilities by controlling the sources from which content can be loaded.

Cobalt discovered the following significant findings during the pentest:

- HTML Injection due to name taking place at Flow
- Insecure file upload and Stored XSS due to crafted SVG file on Application Icon
- Unauthenticated Download of Private Key and Certificate via Direct URL
- Stored XSS via Footer Links
- Weak password policy in use
- Missing CSP Security Header

Cobalt provided specific recommendations for each finding. Overall, the recommendations indicate gaps that could be addressed by implementing secure input validation and improving different misconfigurations in the Application. These improvements include recommendations and security best practices implementations that could be further addressed by the Authentik Security team to increase the organization's overall security posture.

Scope of Work

Target description

Application:

- https://cobalt.pr.test.goauthentik.io/
- http://localhost:9000/

Environment:

- Production

API

- https://cobalt.pr.test.goauthentik.io/api/v3/
- https://localhost:9000/api/v3/

In-scope testing methodologies

Web application

Cobalt performed web application penetration testing according to our Web Application testing methodology. This methodology included a manual assessment of the security of the in-scope asset functionality and business logic, as well as testing for documented vulnerabilities, such as those listed in the OWASP API Top 10 or Common Vulnerabilities and Exposures (CVE) databases.

API

Cobalt performed API penetration testing according to our API testing methodology. This methodology included a manual assessment of the security of the in-scope asset functionality and business logic, as well as testing for documented vulnerabilities, such as those listed in the OWASP API Top 10 or Common Vulnerabilities and Exposures (CVE) databases.

Methodology

The test was done according to penetration testing best practices. The flow from start to finish is listed below.

Pre Engagement

- Scoping
- Customer documentation
- Information discovery

Penetration Testing

- Tool assisted assessment
- Manual assessment
- Exploitation
- Risk analysis
- Reporting

Post Engagement

- Prioritized remediation
- Best practice support
- Retesting

Risk Factors

Each finding is assigned two factors to measure its risk. Factors are measured on a scale of 1 (very low) through 5 (very high).

Impact

This indicates the finding's effect on technical and business operations. It covers aspects such as the confidentiality, integrity, and availability of data or systems; and financial or reputational loss.

Likelihood

This indicates the finding's potential for exploitation. It takes into account aspects such as skill level required of an attacker and relative ease of exploitation.

Severity Definitions

When our pentesters find vulnerabilities, they use the standard OWASP Risk Rating Methodology, and then classify them into one of the following risk levels, based on their business impact and likelihood: `risk = impact * likelihood`

Critical

Includes vulnerabilities that require immediate attention. Risk score of 25.

High

Impacts the security of your application/platform/hardware, including supported systems. Includes high probability vulnerabilities with a high business impact. Risk score range: 16 through 24.

Medium

Includes vulnerabilities that are: medium risk, medium impact; low risk, high impact; high risk, low impact. Risk score range: 5 through 15.

Low

Specifies common vulnerabilities with minimal impact. Risk score range: 2 through 4.

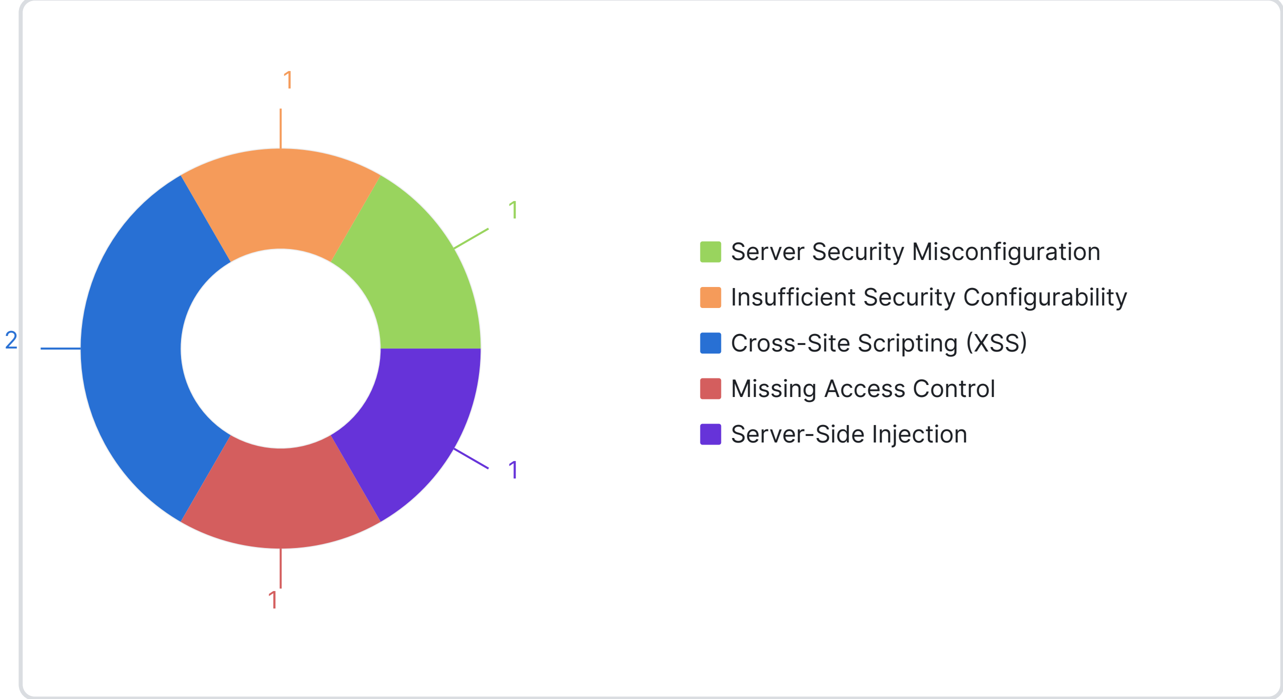
Info

Notes vulnerabilities of minimal risk to your business. Risk score of 1.

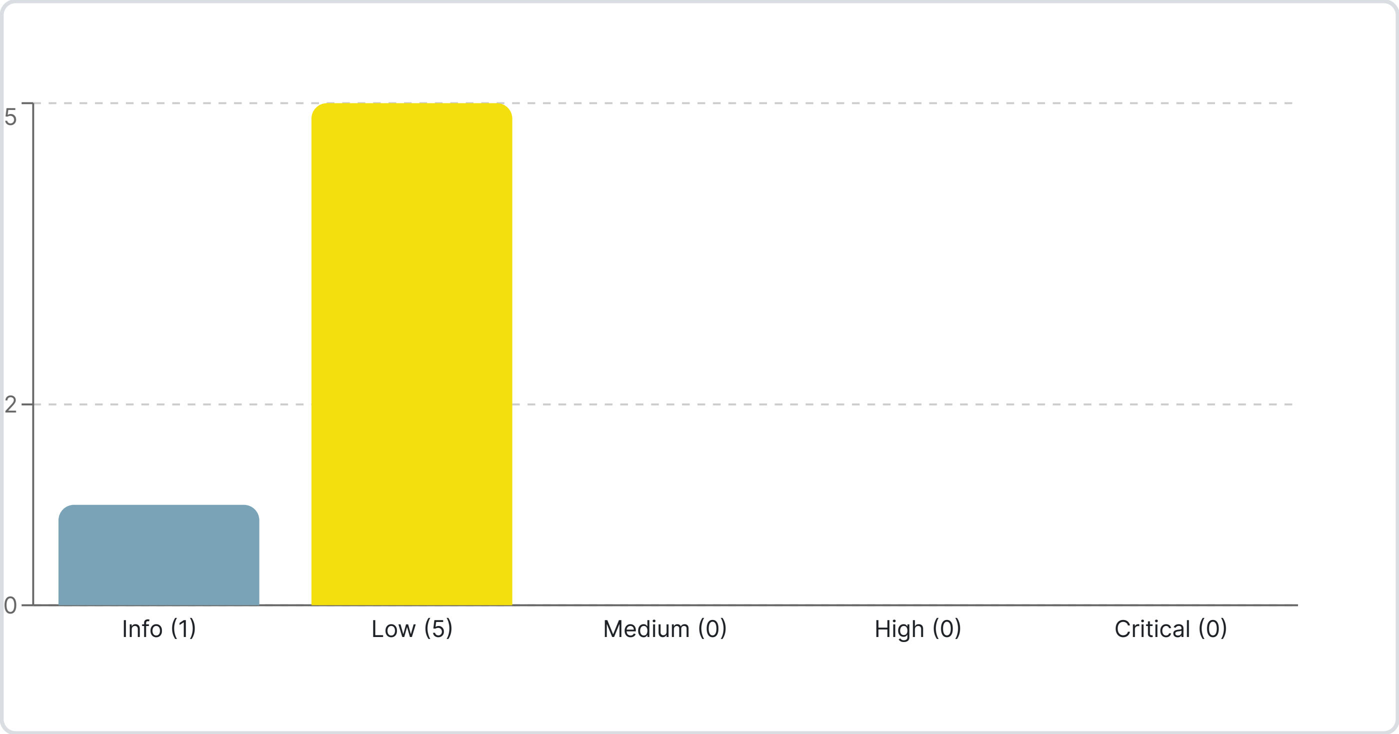
Summary of Findings

The following charts categorize the findings based on the type of finding and the estimated severity.

Findings by Type



Findings by Severity



Analysis

The assessment provided a comprehensive evaluation of the application's security posture. The focus was on analyzing various aspects of the application, including its authentication mechanisms, data handling practices, and overall security configurations. The evaluation aimed to identify potential vulnerabilities and assess the effectiveness of existing security controls. The goal was to ensure that the application adheres to industry standards and best practices, ultimately enhancing its resilience against potential security threats.

After vulnerability analysis, Cobalt reported the following findings:

Low-risk findings

- **HTML Injection due to Name Taking Place at Flow**: During the assessment, it was found that the application was vulnerable to HTML injection through user input in the Flow section. The application did not properly sanitize or encode the HTML content, allowing an attacker to inject and execute arbitrary HTML or JavaScript. (#PT26135_8)

- **Insecure File Upload and Stored XSS due to Crafted SVG File on Application Icon**: The pentesters discovered that the application was vulnerable to both insecure file upload and stored XSS through SVG files. Users could upload SVG files as application icons, but the application failed to adequately validate or sanitize these files. An attacker could upload an SVG file containing malicious scripts, which would then be stored and executed in the context of other users viewing the icon. (#PT26135_7)
- **Stored XSS via Footer Links**: The assessment revealed a stored XSS vulnerability in the application's footer links. The footer section accepted user-provided links without sufficient validation or sanitization, enabling attackers to inject malicious scripts. These scripts are stored and executed when other users interact with the footer links, leading to potential script execution in the users' sessions. (#PT26135_5)
- **Weak Password Policy in Use**: The pentesters found that the application employed a weak password policy, allowing users to create passwords with minimal complexity. This weakness increases the risk of brute-force and dictionary attacks, as the application does not enforce strong password requirements to protect user accounts effectively. (#PT26135_4)
- **Missing CSP Security Header**: It was found that the application did not include a CSP header. The absence of a CSP header means there are no restrictions on the sources of content and scripts the application can load, exposing it to potential XSS attacks and other forms of content injection. (#PT26135_3)

Informational findings

- **Unauthenticated Download of Private Key and Certificate via Direct URL**: It was discovered that sensitive files, including private keys and certificates, were accessible via direct URLs without authentication. This vulnerability allowed unauthorized users to download these files, potentially exposing critical cryptographic information and compromising the application's security.

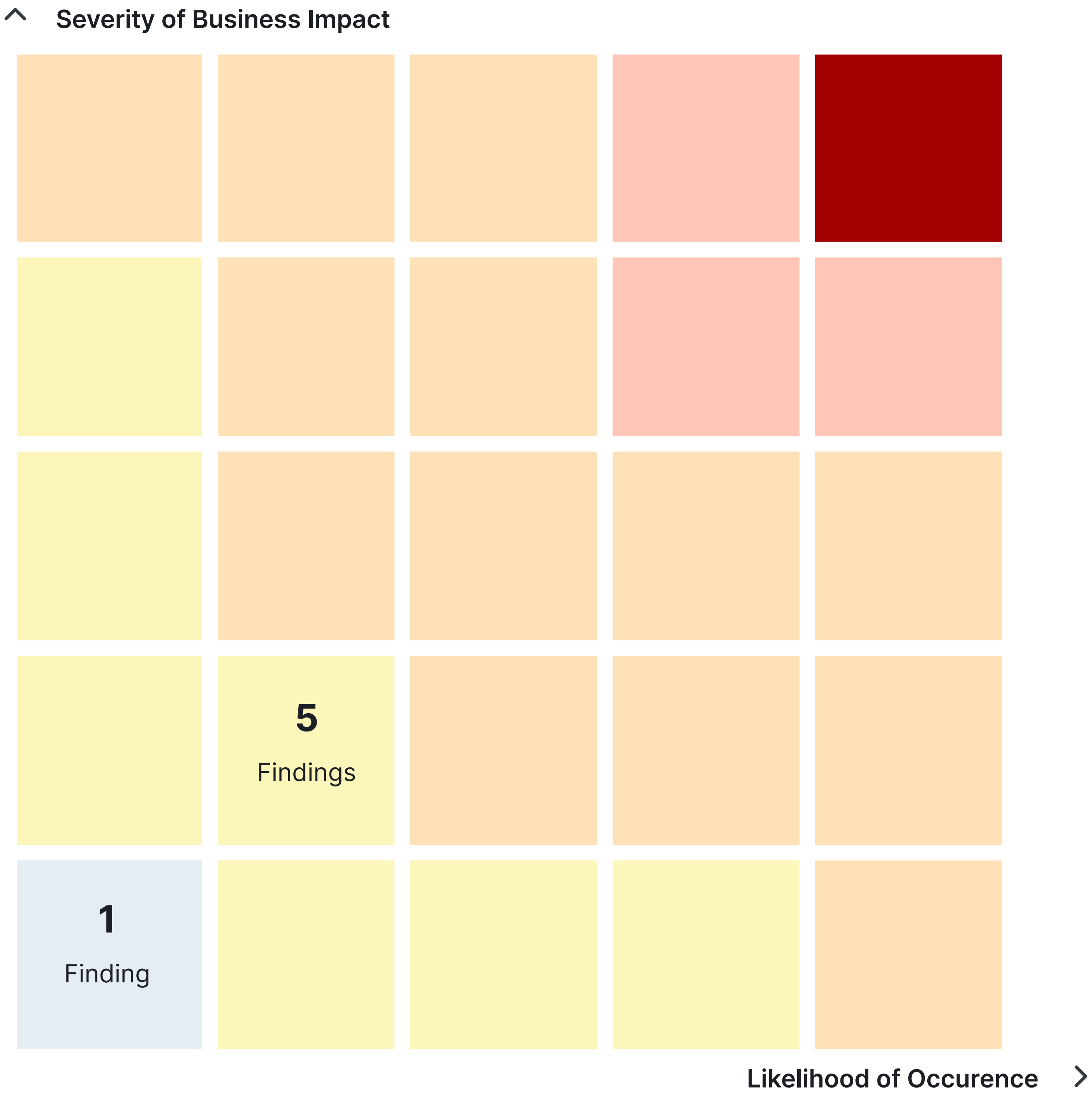
Open ports and services

This section documents all open ports and services identified for the target as well as any potential action needed.

Port	Service and Version	Action needed (if any)
80	http web server	none
443	https web server	none
9000	https web server	none

General Risk Profile

The chart below summarizes vulnerabilities according to business impact and likelihood, increasing to the top right.


Severity of Business Impact
5
Findings
1
Finding
Likelihood of Occurence

Recommendations

As part of an organizational risk management strategy, Cobalt recommends conducting vulnerability assessments regularly. This will allow the organization to determine if newly implemented security controls are correctly installed, operating as intended, and producing the desired outcome. Based on the findings, Cobalt recommends the following remediation:

- Sanitize and encode all user inputs before rendering them in HTML to prevent HTML injection. Implement robust input validation to escape special characters. (#PT26135_8)
- Implement strict validation and sanitization for file uploads, particularly for SVG files. Use a scanning tool to check for and remove any malicious scripts embedded in files. (#PT26135_7)
- Enforce authentication and authorization checks for accessing sensitive files, such as private keys and certificates. Ensure these files are only accessible through secure, authenticated methods. (#PT26135_6)
- Apply validation and sanitization to user-provided content in footer links. Ensure input is checked for malicious content and properly encoded before storage and display. (#PT26135_5)
- Strengthen the password policy by enforcing requirements for minimum length, complexity, and diversity. Require passwords to include uppercase letters, lowercase letters, numbers, and special characters. (#PT26135_4)
- Implement a Content Security Policy (CSP) header to control the sources of content and scripts that the application can load. Define CSP rules to restrict content sources and mitigate cross-site scripting (XSS) and content injection attacks. (#PT26135_3)

Post-Test Remediation

All identified findings are below with their mitigation status.

Finding	Type	Severity	State	Resolved
#PT26135_3	Server Security Misconfiguration	●LOW	Fixed	28 Nov 2024
#PT26135_4	Insufficient Security Configurability	●LOW	Fixed	28 Nov 2024
#PT26135_5	Cross-Site Scripting (XSS)	●LOW	Fixed	28 Nov 2024
#PT26135_7	Cross-Site Scripting (XSS)	●LOW	Fixed	28 Nov 2024
#PT26135_8	Server-Side Injection	●LOW	Fixed	28 Nov 2024
#PT26135_6	Missing Access Control	●INFO	Fixed	28 Nov 2024

Terms

PLEASE NOTE: It is impossible to test networks, information systems, and people for every potential security vulnerability. This report does not form a guarantee that your assets/targets are secured from any and all threats. All assessments performed, and their results, are only from the point-of-view of Cobalt, at the time of the engagement. Cobalt is unable to ensure or guarantee that your assets/targets are or will be completely safe from every form of attack now or in the future. With the ever-changing environment of information technology, any assessment performed by Cobalt will necessarily exclude vulnerabilities in software or systems that are unknown at the time of the engagement. For a full list of terms governing the services of Cobalt, this report, and the usage thereof, please consult the Terms of your Agreement with Cobalt or www.cobalt.io/terms.