



Project number: 0058400
Published: 2022-05-30
Document name: Testbericht Penetrationstest GAIA-X._V1.2docx.docx
Document version: 1.2
Classification: Confidential
Test procedure: Verfahrensanweisung Penetrationstest, Version 3.0
For: Helmi Ben Hmida
T-Systems International GmbH

Basis for test: (according to)
ISO/IEC 25051: 2014
ISO/IEC 25010: 2011



LIFE IS FOR SHARING.

Imprint

Publisher

Test and Integration Center
T-Systems Multimedia Solutions GmbH
Riesaer Str. 5
01129 Dresden

Rights are reserved, including partial reprints, photo-mechanical reproduction (including micro-copy) as well as the evaluation using databases or similar means.

The Test and Integration Center Dresden of T-Systems Multimedia Solutions GmbH is a test laboratory accredited by the DAkkS according to DIN EN ISO/IEC 17025:2018. The accreditation applies to the test procedures listed in the certificate.

Register number of the certificate: **D-PL-12109-01-00**



This document is part of the accredited test procedure Software engineering - Software product evaluation - Quality requirements for software products (ISO/IEC 25051:2014).

Document Information

Title	Customer
Penetration Test - GAIA-X	T-Systems International GmbH
Author	Project Manager
Patrick Walker	Antje Winkler
Document version	Document status
1.2	Final

This document describes the summarized Penetration Test results of GAIA-X performed by Test and Integration Center (TIC) - T-Systems Multimedia Solutions GmbH.

This test report may only be distributed as a whole (except with the written consent of the TIC).

Changelog

Version	Date	Editor	Changes / Comments
0.1	2022-05-24	Patrick Walker	Initial draft
0.2	2022-05-25	Patrick Walker	First results
0.3	2022-05-30	Patrick Walker	Finalization of the report
1.0	2022-05-30	Antje Winkler	Review and release
1.1	2022-06-02	Antje Winkler	Reworked based on customer feedback
1.2	2022-06-02	Antje Winkler	Added further endpoints in scope

Contact Information

Name	Telephone number	E-Mail-Address	Role
Patrick Walker	+49 351 2820 2042	Patrick.Walker@t-systems.com	Tester
Antje Winkler	+49 351 2820 2093	Antje.Winkler@t-systems.com	Test Manager

Distribution List

Name	Function/Company
Helmi Ben Hmida	Chapter Large Scale Prog. Mgmt. / T-Systems International GmbH

Release

Name	Position	Signature
Antje Winkler	Project Manager / Test Manager	-----

The report in electronic form is valid with a digital signature.

Table of Contents

1	Overall Summary	6
1.1	Test Objectives	6
1.2	Test Implementation Details.....	6
1.3	Management Summary	7
1.4	Evaluation of Results	7
1.5	Overview Test Results.....	8
2	General	9
2.1	Keys Aspects of Testing	9
2.2	Test Limitations	10
2.3	List of Provided Documents.....	10
2.4	Test Tools	10
2.5	General Information	11
2.5.1	Evaluation of Results	11
2.5.2	Description of Severity Levels	12
3	Explanation of the Test Procedure.....	13
3.1	Preparation	13
3.2	Reconnaissance	14
3.3	Analyzing Information and Risks.....	14
3.4	Active Intrusion Attempts.....	14
3.5	Final Analysis / Rework / Clean-up	15
4	Test Results.....	16
4.1	REST API	16
4.1.1	Cryptography	16
4.1.2	Authentication Testing.....	17
4.1.3	Authorization Testing.....	17
4.1.4	Configuration and Deployment Management Testing	19
4.1.5	Input Validation Testing.....	19
4.1.6	Business Logic Testing.....	19
4.1.7	Client-Side Testing.....	19
4.1.8	Error Handling	20
4.2	Negativ-Test SpringBoot Actuator-Endpoints	20
5	Appendix.....	23
5.1	Description of Severity Levels	23
5.1.1	Severity Level „Informational“	23
5.1.2	Severity Level „Low“	23
5.1.3	Severity Level „Medium“	24
5.1.4	Severity Level „High“	25
5.1.5	Severity Level „Critical“	25

List of Tables

Table 1: Overall Test Results	8
Table 2: Key aspects of testing	10
Table 3: List of provided documents.....	10
Table 4: Test Tools	11
Table 5: Criteria of the procedure and methodology in the preparation phase.....	13

List of Figures

Figure 1: First authorization request, redirecting to the login form	18
Figure 2: Login form with pre-filled, hidden username and password fields.....	18
Figure 3: Login submission with redirect to authorization endpoint and finally to target application .	19
Figure 4: Heapdump Actuator Endpoint enabled	21
Figure 5: Private key data found in actuator heapdump.....	22

1 Overall Summary

1.1 Test Objectives

Project Name	GAIA-X
Short Description of Test Object	This is a Reference Implementation of GAIA-X LOT1 Authentication & Authorization Service.
Type of Test Object	Web Services
URL/IP Address/Test Object	<p>Standard Auth Endpoints for OpenID Connect:</p> <ul style="list-style-type: none">▪ /.well-known/openid-configuration▪ /oauth2/jwks▪ /oauth2/authorize▪ /oauth2/token▪ /userinfo <p>Custom endpoints:</p> <ul style="list-style-type: none">▪ /ssi/login▪ /ssi/qr/{requestId}▪ /clients/iat/requests▪ /clients/iat/requests/{{request_id}}
Deployment Environment	Testing Environment
Version Number of Test Object	1.0
Contact Person	Helmi Ben Hmida, Helmi.Ben-Hmida@t-systems.com Denis Sukhoroslov, Denis.Sukhoroslov@t-systems.com Ladislav Juncisin, Ladislav.Juncisin@external.t-systems.com

1.2 Test Implementation Details

Test Type	Penetration Test of Interfaces / Web Services
Test Period	2022-05-23 to 2022-05-25
Contact Details of the Tester	Patrick Walker; +49 351 - 2820 2024; Patrick.Walker@t-systems.com
Test Requirements and Provisions	<ul style="list-style-type: none">▪ API description (e.g. as postman collection)▪ Valid user credentials for accessing the service
Test Laboratory	Test and Integration Center T-Systems Multimedia Solutions GmbH Riesaer Str. 5 01129 Dresden

1.3 Management Summary

In the penetration test, a technical security examination of the service is carried out. The procedure of the analysis is based on the penetration tests implementation concept of the Federal Office for Information Security (BSI) to identify potential weaknesses.

During the penetration test of the GAIA-X Authorization and Authentication service, 2 issues have been identified, both of which are rated as having a high severity. The tested application did not require any authentication. Instead, a login form with a pre-filled claim token (in the hidden username field) that only offers a "Login" button gets sent by the server, once the OAuth2 authorization request determines that the user is currently not logged in. This form then needs to be sent back to the server and automatically logs the user into the target application. Since there are no actual credentials required, the authentication has to be seen as non-existent.

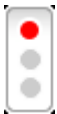
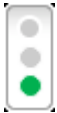
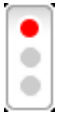
Additionally, the Spring Boot Actuator endpoints were accessible without any authentication. These endpoints can be used to obtain confidential information and should thus be protected from unauthorized access. During the test, it was possible to retrieve a secret private RSA key that is used for signing JSON Web Tokens (which, in turn, are used for authenticating against (other) applications).

During the audit, the test environment was only accessible via unencrypted HTTP. For production deployments, the application should be properly secured with a TLS tunnel (i. e. via HTTPS). The configuration of the TLS layer should follow recommendations of the Federal Office for Information Security (BSI).

An overview of the documented vulnerabilities is given in chapter 1.5. The detailed test results can be found in chapter 4.

1.4 Evaluation of Results

Based on predefined criteria and the results of the penetration test performed, the following chapter assesses the current security level of the application and concludes with an overall evaluation. A description of the evaluation criteria can be found in chapter 2.5.

Evaluation Criteria	Rating	Reason
Vulnerabilities		High or even critical vulnerabilities were identified.
Test Coverage		All aspects of the defined test scope were inspected
Overall Rating		Based on the number and severity of identified vulnerabilities and the test coverage level, an overall security risk rating of High was chosen by the testing contractor.

1.5 Overview Test Results

No.	Chapter	Description	Severity	Status as of 2022-05-30
1	4.2.1.1	Spring Boot Actuator Endpoints available	High	Open
2	4.1.3.1	Access Control - Missing or Incorrect Authentication	High	Open
3	4.1.1.1	Cleartext Transmission of Sensitive Information	Informational	Open

Table 1: Overall Test Results

2 General

Within the framework of the penetration test, the application is tested for existing vulnerabilities from an attacker's perspective, which provides a security assessment level. In particular, a penetration test provides an overview of identified vulnerabilities that could be exploited to compromise the confidentiality, integrity, and availability of the application or IT system.

The following chapter summarizes the details of test methodology. Chapter 3 describes the general procedure of a penetration test. For each identified vulnerability, chapter 4 contains a description, metadata, evidence and recommended actions.

2.1 Keys Aspects of Testing

In terms of content, the main focus of testing is based on best practices and official test guidelines (e.g. OWASP Testing Guide, BSI study "Implementation Concept for Penetration Testing", ATT&CK™, etc.). In the penetration test, the key aspects of testing based on OWASP Testing Guides in version 4 were used.

System Component	Key Aspects of Testing
REST API <ul style="list-style-type: none">Standard OIDC endpointsCustom endpoints	<ul style="list-style-type: none">Cryptography, e.g.<ul style="list-style-type: none">Check of transport encryption and the validity of the used certificateAuthentication Testing, e.g.<ul style="list-style-type: none">Test of correct validation of API keys and tokens (e.g., JWT)Test for trust relationships to IdPsTest of omitting authentication mechanismsAuthorization Testing, e.g.<ul style="list-style-type: none">Non-public REST servicesMissing or insufficient authorization checksConfiguration and Deployment Management Testing, e.g.<ul style="list-style-type: none">Testing of supported HTTP methodsTesting of outdated software versionsTesting for availability of administrative interfacesTesting for unneeded servicesTesting for presence of HTTP Security HeadersInput Validation Testing, e.g.<ul style="list-style-type: none">Test of weaknesses in the used parsersTesting for various injection possibilitiesTesting for user-controlled resource allocation (Denial of Service)Test for handling of nested and reoccurring objectsBusiness Logic Testing, e.g.<ul style="list-style-type: none">Check for weaknesses in business logic and processesTest for attack vectors that could affect the application platform availability (input-dependent resource allocation)Client-Side Testing, e.g.<ul style="list-style-type: none">Cross Origin Resource Sharing (CORS)Error Handling, e.g.<ul style="list-style-type: none">Test for information disclosure

Negativ-Test

- Check for accessibility of management interfaces

SpringBoot Actuator-
endpoints

Table 2: Key aspects of testing

2.2 Test Limitations

All tests could be performed without any limitations.

2.3 List of Provided Documents

Relevant information or documents were provided by the project team.

Table 3 summarizes the documents and information provided in advance by the customer for testing.

Document type	Filename	Version, State
Postman collection of OIDC Endpoints	pen_testing.postman_collection.json	2022-05-20
Security Testing Requirements	SecurityTesting_draft_v1.docx	2022-05-20
OIDCC-Server Test Summary	test-log-oidcc-server-HBO6NL9MZCSNM6v.html	2022-05-20
OpenAPI documentation of IAT provider	iat_provider.yml	2022-05-20

Table 3: List of provided documents

2.4 Test Tools

The following tools were used during the test:

No.	Tool name	Application area	Manufacturer
1	Quality Center	Documentation of test cases Evaluations Compliance Management	HP
2	MS Word	various documentation, reports	Microsoft
3	MS EXCEL	various documentation	Microsoft
4	Share Point	Version control system for documentation (reports, concepts)	Microsoft
5	Burp Suite Pro	Dynamic web application tests	Portswigger
6	Firefox	Dynamic web application tests	Mozilla Foundation

7	Postman	Dynamic web API tests	Postman, Inc.
8	binwalk	Data mining and carving	ReFirm Labs
9	openssl	Collection of cryptographic tools	The OpenSSL Project

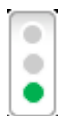
Table 4: Test Tools

2.5 General Information

2.5.1 Evaluation of Results

The evaluation of results in the chapter 1.4 is based on the following criteria:

Evaluation Criteria – Vulnerabilities



There were no vulnerabilities or vulnerabilities with the severity of informational and low identified.



At least one vulnerability has been identified with the severity of medium, but no high or critical vulnerabilities were discovered.



At least one vulnerability with the severity of high has been identified.

Evaluation criteria – Test Coverage

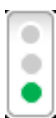
The test coverage criteria signify the test scope that have been tested concerning planned test scope. Remaining untested components may contain further unknown vulnerabilities that might lead to residual risks.

During the test, it may not be possible to test all components mentioned in the scope. There are several reasons for this. A few reasons are listed below.

- For example, some parts of the software were not ready for the testing, or the test requirements have not been met.
- In the case of complex applications or IT infrastructures, the subjective impression may arise, for example, due to time constraints, that a complete test could not be carried out within the agreed test period.

The ratio of the tested scope to the planned or expected test scope is represented as test coverage.

During the test, there could be areas and functions that go beyond the planned test scope. In that case of incomplete test coverage, there is a residual risk, since the untested components may contain further unknown vulnerabilities.



(Approximately) All the components mentioned in the planned test scope were tested.



Not all components of the test scope were tested. However, it can be easily estimated that the residual risk is low based on the components tested.



It was not possible to test all the components mentioned in the test scope. There is a high probability that the untested parts of the application or IT infrastructure may contain further unknown vulnerabilities.

2.5.2 Description of Severity Levels

The method used in this document to determine the severity of vulnerabilities found is based on the **CVSS approach**. CVSS stands for Common Vulnerability Scoring System and is an industry-standard for calculating the severity of vulnerabilities (<https://www.first.org/cvss/>).

Severity levels are calculated based on identifying the characteristic of vulnerability that have a risk impact. These characteristics are further divided into two factors namely, probability of occurrence (likelihood) and impact.

Only the CVSSv3 Base Score is used to determine the severity level.

The numerical values (0.0 to 10.0) result in the following severity levels.

CVSSv3 Base Score	Severity Level
0.0	Informational
0.1 - 3.9	Low
4.0 - 6.9	Medium
7.0 - 8.9	High
9.0 - 10.0	Critical

3 Explanation of the Test Procedure

The procedure model is based on the "A Penetration Testing Model"¹ published by the Federal Office for Information Security and is divided into 5 individual phases.

3.1 Preparation

The first stage is carried out in close coordination with the client. This phase aims to determine the scope of the test, the general conditions, and risk factors.

The procedure of the penetration tester depends mainly on the following criteria.

Information basis	What is the penetration tester's initial level of knowledge (Blackbox/Whitebox) about the target (network or object)?
Aggressiveness	How aggressive should be the penetration tester during the test (passive/cautious/balanced/aggressive)?
Scope	Which systems are to be tested (definition/boundary/focus)?
Approach	How "visible" should the team proceed while testing (Covert (stealthy)/Overt (noisy))?
Technique	Which techniques should be used for testing (network-based/ other communication/physical access/ social engineering)?
Starting point	From which perspective does the penetration test should perform(outside/inside perspective)?

Table 5: Criteria of the procedure and methodology in the preparation phase

Based on the results of the preparatory phase, the final test priorities are determined (see Chapter 2.1). The test priorities replace the modules for information gathering and active intrusion tests (I and E modules) listed in the BSI implementation concept. However, the BSI modules are only applicable to selected systems/web applications, so that T-Systems Multimedia Solutions GmbH has developed its own test modules and associated test cases for various system components and test perspectives.

The potential risks that might occur during the test are discussed with the client. The technical risks are reduced since the penetration testers have extensive experience, but cannot be completely excluded. Nevertheless, fragile test components, cost- or resource-intensive process steps (e.g. SMS dispatch) or similar are discussed and agreed with the client. In the interest of both parties, all measures are to be taken to keep the risks posed by penetration testing to a minimum level.

The contact possibilities during the test period must be recorded bilaterally with the client. This procedure enables immediate contact in case of technical problems or queries.

The results of the preparation phase were documented in detail in Chapter 1.

¹ https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/Penetration/penetration_pdf.html

3.2 Reconnaissance

Depending on the type of test object, various information about the target is collected in order to perform the test in a more effective way.

- In the case of an IT infrastructure, for example, the communicated network addresses are examined more intensively and information on associated (sub-) domains is collected.
- In the case of (web) applications, the technologies used are checked and any available verification data of the software used is collected. For the identified software versions, existing vulnerabilities or the existence of public exploits are researched.
- When investigating hardware components, the available interfaces are analyzed and relevant test cases are derived.

Depending on the information collected, test cases that were previously marked as relevant can be deselected or the scope of test cases can be expanded if additional attack vectors or technologies are identified at this stage. The concrete compilation of the test cases is carried out by the tester individually or, if required, with the contact person on the customer side.

Besides, this phase enables the tester to assess the scope of the test more accurately and to plan the originally planned time expenditure per component/target more precisely.

3.3 Analyzing Information and Risks

An elementary task of the third phase "Analyzing information" is the professional and technical analysis of the test object in order to identify and evaluate threats.

Based on this information, it is possible to assign appropriate severity levels for the application or the IT infrastructure at a later stage and to make a realistic assessment of the potential threat.

It is therefore necessary for the tester to analyze the information provided (e.g. documents from the department, analysis of the application, research on the technologies used, possibly questioning the department) and to derive from this which goals a potential attacker could pursue in order to prioritize the test cases and to concretely define the focus of the penetration test. In the case of more complex systems and applications, this phase can only be passed through to a limited extent due to time constraints and possibly the large volume of information received.

Based on the functional and technical possibilities of the test object, the tester assesses the chances of success of a target-oriented attack in order to make a decision on the test depth and intensity (e.g. time expenditure) of selected functions and the defined targets. For example, some functions have a higher exploitation potential (e.g. write accesses such as file uploads) and should therefore have a higher priority in the test period.

Based on the experience and specialization of the tester, the result of this phase will be subjective.

3.4 Active Intrusion Attempts

According to the agreements from the preparation phase (see Chapter 3.1), the tester carries out appropriate test cases based on the test priorities.

For each identified (potential) vulnerability:

- existing **boundary conditions** are examined (e.g. necessary user rights, existing protection mechanisms, conditions for exploitation),

- **evidences** are generated (screenshots, commands used, navigation in the application towards the relevant function, code snippets, requests and responses of an attack, proof of concepts, etc.) to facilitate comprehensibility and reproducibility,
- identifies the potential **impact** of the vulnerability, and
- if necessary, any **further vulnerabilities** are discovered (e.g. to create a remote shell from an SQL injection in order to execute system commands).

In this phase, new findings/information are collected if necessary, so that the two phases „Reconnaissance“ and „Analyzing Information and Risks “ are passed through iteratively.

3.5 Final Analysis / Rework / Clean-up

The results of the „Active Intrusion Attempts“ phase are collected, evaluated and documented in detail in the final test report. The aim of the documentation is to provide the client and the parties involved with all the necessary information in a detailed and comprehensible manner in order to highlight relevant risks for the purpose of monitoring business operations.

Therefore, the test report contains a meaningful summary for the management, which includes the essential test results and recommended further procedures on an abstract level. However, the main part of the final report is a detailed description and evaluation of the test results. In the final analysis, each vulnerability is described in detail and its effects and possible remedies are explained. The recommended measures for the documented vulnerabilities are generally formulated in concrete terms, but the final decision on the type and scope of remedy(s) is the responsibility of the client.

It may be necessary to re-examine the framework conditions of a vulnerability during documentation, e.g. if a new attack scenario arises in combination with another vulnerability.

Sensitive data recorded during the test (e.g. personal data or access data of other persons) will be securely deleted if no longer needed and will be made unrecognizable in the final documentation. Traces of the test (e.g. backdoors) must also be removed from the systems, especially if they could be used for misuse or further attacks by others. If the penetration test has taken place in a dedicated test environment, which is completely removed or cleaned up after the end of the test, the final cleanup can be omitted.

4 Test Results

4.1 REST API

4.1.1 Cryptography

4.1.1.1 Cleartext Transmission of Sensitive Information

Description

If a service, an application, or an IT system transmits sensitive or security-critical information through a plain-text communication channel such as HTTP, this information is not protected against eavesdropping and manipulation. Unencrypted communication and transmission of information increases the risk for the application as well as for its users, since an attacker can read and manipulate the data on the communication channel using a man-in-the-middle attack.

Result

The test setup was only available via unencrypted HTTP, but not via encrypted HTTPS.

Since this was just a testing environment, this finding is rated as "Informational". For production environments, this would have resulted in a high severity rating.

Impact

An attacker can potentially - undetected - position himself between the communication partners and read and manipulate transmitted data on the communication channel (man-in-the-middle attack). An attacker can use gained information to plan and perform further attacks.

Furthermore, the integrity and confidentiality of the transmitted data are not sufficiently guaranteed.

Affected System / Host / Function

- <http://78.138.66.89:9000/>

References

- https://www.owasp.org/index.php/Insecure_Transport
- https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/09-Testing_for_Weak_Cryptography/03-Testing_for_Sensitive_Information_Sent_via_Unencrypted_Channels
- https://wiki.mozilla.org/Security/Server_Side_TLS

Severity: Informational

CVSSv3 Base Score: 0.0 (Vector CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:N/I:N/A:N)

Recommendation

The following points should be considered when transmitting sensitive and security-critical data:

- Use of HTTPS (HTTP with SSL / TLS) for the transmission of data to ensure appropriate transport encryption
- The transport encryption should be based on TLSv1.2 (or higher)

- Used cipher suits should support Perfect Forward Secrecy (PFS), that means a new secret key is negotiated for each TLS session
- A valid certificate has to be used so that an active man-in-the-middle attack can be identified by users (certificate warning)
- On server-side the HTTP Strict Transport Security Header (HSTS) should be set to enforce the use of HTTPS on the client side

It is not enough to encode sensitive or security-critical data (e.g. Base64) and to transmit it over an unencrypted communication channel, because an attacker can restore the plain text with a little effort. Even if the communication protocol is not human-readable but binary, encryption is needed.

4.1.2 Authentication Testing

Test passed without any security issues.

4.1.3 Authorization Testing

4.1.3.1 Access Control - Missing or Incorrect Authentication

Description

The system or application does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

Access control involves the use of several protection mechanisms such as:

- Authentication (proving the identity of an actor) and
- Authorization (ensuring that a given actor can access a resource).

Authentication is providing and validating identity of a user.

In case of **missing authentication**, the system does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources.

Incorrect authentication describes the behavior of a system or application when an attacker claims to have a given identity, but the application does not prove or insufficiently proves that the claim is correct.

Result

Starting the OIDC workflow by accessing the authorization endpoint, users get redirected to the `/ssi/login` endpoint which contains a server-generated claim, placed inside the hidden `username` field. The `password` field contains the value `OIDC`, indicating that the OpenID Connect authentication type should be used. The user then only needs to press the `Login` button (or scan the QR code of the claim with an undefined mobile application) and automatically gets logged into the target application.

At no point in that flow, actual credentials are requested from the user (and no prior explicit authentication had taken place), so in essence, there is no real authentication of the user.

Impact

In consequence, if an authentication check is not applied or otherwise fails, attackers can compromise the security of the software by gaining privileges, reading sensitive information, executing commands, evading detection, etc.

Furthermore, exposing critical or administrative functionality essentially provides an attacker with the privilege level of that functionality. The consequences will depend on the associated functionality, but they can range from reading or modifying sensitive data, access to administrative or other privileged functionality, or possibly even execution of arbitrary code.

Affected System / Host / Function

- <http://78.138.66.89:9000/>

Screenshot

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP
279	https://localhost.emobix.co.uk:8443	GET	/test/a/aas-test/callback?code=W10kG...	✓								✓	127.0.0.1
278	http://78.138.66.89:9000	GET	/oauth2/authorize?client_id=aas-app-...	✓		302	497						78.138.66.89
277	http://78.138.66.89:9000	POST	/login	✓		302	565						78.138.66.89
272	http://78.138.66.89:9000	GET	/ssi/login			200	36749	HTML		Login			78.138.66.89
271	http://78.138.66.89:9000	GET	/oauth2/authorize?client_id=aas-app-...	✓		302	320						78.138.66.89

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre>1 GET /oauth2/authorize?client_id=aas-app-oidc&redirect_uri= https://localhost.emobix.co.uk:8443/test/a/aas-test/callback&scope=openid&state= kvk8QoJolN&nonce=ev1IG3XNR&response_type=code HTTP/1.1 2 Host: 78.138.66.89:9000 3 Upgrade-Insecure-Requests: 1 4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.67 Safari/537.36 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng ,/*;q=0.8,application/signed-exchange;v=b3;q=0.9 6 Accept-Encoding: gzip, deflate 7 Accept-Language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7 8 Cookie: JSESSIONID=5A428D32D168C174A61CE78797F53A43 9 Connection: close</pre>				<pre>1 HTTP/1.1 302 2 X-Content-Type-Options: nosniff 3 X-XSS-Protection: 1; mode=block 4 Cache-Control: no-cache, no-store, max-age=0, must-revalidate 5 Pragma: no-cache 6 Expires: 0 7 X-Frame-Options: DENY 8 Location: http://78.138.66.89:9000/ssi/login 9 Content-Length: 0 10 Date: Mon, 30 May 2022 07:25:08 GMT 11 Connection: close</pre>			

Figure 1: First authorization request, redirecting to the login form

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP
272	http://78.138.66.89:9000	GET	/ssi/login			200	36749	HTML		Login			78.138.66.89

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre>1 GET /ssi/login HTTP/1.1 2 Host: 78.138.66.89:9000 3 Upgrade-Insecure-Requests: 1 4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.67 Safari/537.36 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a png,/*;q=0.8,application/signed-exchange;v=b3;q=0.9 6 Accept-Encoding: gzip, deflate 7 Accept-Language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7 8 Cookie: JSESSIONID=5A428D32D168C174A61CE78797F53A43 9 Connection: close</pre>				<pre>94 </p> 95 </div> 96 </header> 97 <main> 98 <div class="card"> 99 <div class="card-top"> 100 <h1> Welcome to Gaia-X </h1> 101 <p> Sign in to continue </p> 102 </div> 103 <hr class="card-divider"> 104 <div class="card-bottom"> 105 <div class="card-bottom-content"> 106 <h2 id="greeting"> Scan the QR code with your mobile device 107 </h2> 108 <img src= /ssi/qr/dXJpOisvYTI1OWESNTgcOTNmYS00M2UzLW13YTctZjdiMmM3ODdkMGUS" 109 alt="QR Code" 110 <form action="/login" method="post"> 111 <table> <tr> <td> <input class="input-box" type="hidden" name="username" value="a9e9a550-53fa-47e3-b7a7-f7b2c7b7a0e9"/> </td> </tr> <tr> <td> <input class="input-box" type="hidden" name="password" value="OIDC"/> </td> </tr> <tr> <td> <input id="sign-in-button" class="action_button" name=" qr-smartphone-button" type="submit" 112 </td> 113 </tr> 114 </tr> 115 </table> 116 </form> 117 </div></pre>			

Figure 2: Login form with pre-filled, hidden username and password fields

279	https://localhost.emobix.co.uk:8443	GET	/test/aas-test/callback?code=W10kG...	✓					✓	127.0.0.1
278	http://78.138.66.89:9000	GET	/oauth2/authorize?client_id=aas-app...	✓	302	497				78.138.66.89
277	http://78.138.66.89:9000	POST	/login	✓	302	565				78.138.66.89

Request		Response	
Pretty	Raw	Pretty	Raw
<pre> 1 POST /login HTTP/1.1 2 Host: 78.138.66.89:9000 3 Content-Length: 86 4 Cache-Control: max-age=0 5 Upgrade-Insecure-Requests: 1 6 Origin: http://78.138.66.89:9000 7 Content-Type: application/x-www-form-urlencoded 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.67 Safari/537.36 9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9 10 Referer: http://78.138.66.89:9000/ssi/login 11 Accept-Encoding: gzip, deflate 12 Accept-Language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7 13 Cookie: JSESSIONID=545209320160C174A61CE78797F53A93 14 Connection: close 15 16 username=a5e9a958-93fa-47e3-b7a7-f7b2c787d0e9&password=01DC4qr-smartphone-button= Login </pre>		<pre> 1 HTTP/1.1 302 2 Set-Cookie: JSESSIONID=9C4E3EB16EBD6B5CEB117B33CD6AF811; Path=/; HttpOnly 3 X-Content-Type-Options: nosniff 4 X-XSS-Protection: 1; mode=block 5 Cache-Control: no-cache, no-store, max-age=0, must-revalidate 6 Pragma: no-cache 7 Expires: 0 8 X-Frame-Options: DENY 9 Location: http://78.138.66.89:9000/oauth2/authorize?client_id=aas-app-oidc&redirect_uri=http://localhost.emobix.co.uk:8443/test/aas-test/callback&scope=openid&state=kwkBqoJcINAnonce=ev1UG3XMP&response_type=code 10 Content-Length: 0 11 Date: Mon, 30 May 2022 07:27:19 GMT 12 Connection: close 13 14 </pre>	

Figure 3: Login submission with redirect to authorization endpoint and finally to target application

References

- https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet
- https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet
- https://cheatsheetseries.owasp.org/cheatsheets/Transaction_Authorization_Cheat_Sheet.html
- https://cheatsheetseries.owasp.org/cheatsheets/Access_Control_Cheat_Sheet.html
- https://owasp.org/Top10/A01_2021-Broken_Access_Control/

CVSSv3 Base Score: 7.2 (Vector CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:N)

Recommendation

Implement a proper authentication flow.

4.1.4 Configuration and Deployment Management Testing

Test passed without any security issues.

4.1.5 Input Validation Testing

Test passed without any security issues.

4.1.6 Business Logic Testing

Test passed without any security issues.

4.1.7 Client-Side Testing

Test passed without any security issues.

4.1.8 Error Handling

Test passed without any security issues.

4.2 Negativ-Test SpringBoot Actuator-Endpoints

4.2.1.1 Spring Boot Actuator Endpoints available

General Description

Spring Boot includes a number of additional features to help developers to monitor and manage their application when pushing it to production. Developers can choose to manage and monitor the application by using HTTP endpoints or with JMX. Auditing, health, and metrics gathering can also be automatically applied to the application. The `spring-boot-actuator` module provides all of Spring Boot's production-ready features.

Since these endpoints expose sensitive data (e. g. configuration and environment properties), access to the actuator endpoints needs to be restricted. Additionally, there are endpoints for dumping heap and/or thread memory which in turn can contain further sensitive information (e. g. private keys, passwords, etc.).

Result

During the test, it was found that the Spring Boot actuator endpoints are accessible without any authentication. Due to that, a dump of the heap memory could be requested, while also performing business-critical actions (e. g. attempting to log into the application).

Afterwards, the heap dump file was investigated, and a private RSA key could be extracted. It turned out that this key is used for signing JWT, and would thus allow to forge arbitrary, valid JSON Web Tokens, compromising the whole authentication and authorization workflow.

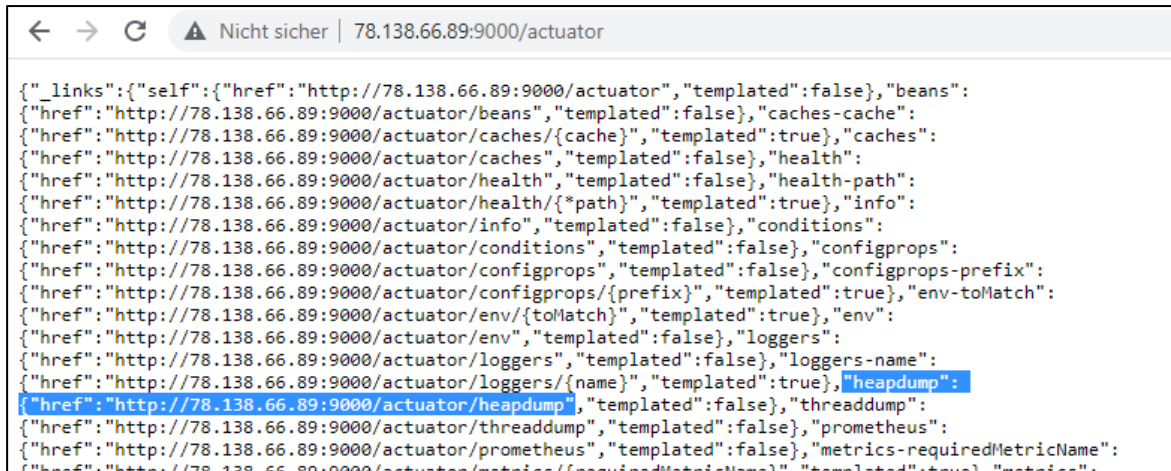
Impact

Attackers with access to the Spring Boot actuator endpoints may get access to sensitive information.

Affected System / Host / Function

- `http://78.138.66.89:9000/actuator`

Screenshot



```
{
  "_links": {
    "self": {
      "href": "http://78.138.66.89:9000/actuator",
      "templated": false
    },
    "beans": {
      "href": "http://78.138.66.89:9000/actuator/beans",
      "templated": false
    },
    "caches-cache": {
      "href": "http://78.138.66.89:9000/actuator/caches/{cache}",
      "templated": true
    },
    "caches": {
      "href": "http://78.138.66.89:9000/actuator/caches",
      "templated": false
    },
    "health": {
      "href": "http://78.138.66.89:9000/actuator/health",
      "templated": false
    },
    "health-path": {
      "href": "http://78.138.66.89:9000/actuator/health/{*path}",
      "templated": true
    },
    "info": {
      "href": "http://78.138.66.89:9000/actuator/info",
      "templated": false
    },
    "conditions": {
      "href": "http://78.138.66.89:9000/actuator/conditions",
      "templated": false
    },
    "configprops": {
      "href": "http://78.138.66.89:9000/actuator/configprops",
      "templated": false
    },
    "configprops-prefix": {
      "href": "http://78.138.66.89:9000/actuator/configprops/{prefix}",
      "templated": true
    },
    "env-toMatch": {
      "href": "http://78.138.66.89:9000/actuator/env/{toMatch}",
      "templated": true
    },
    "env": {
      "href": "http://78.138.66.89:9000/actuator/env",
      "templated": false
    },
    "loggers": {
      "href": "http://78.138.66.89:9000/actuator/loggers",
      "templated": false
    },
    "loggers-name": {
      "href": "http://78.138.66.89:9000/actuator/loggers/{name}",
      "templated": true
    },
    "heapdump": {
      "href": "http://78.138.66.89:9000/actuator/heapdump",
      "templated": false
    },
    "threaddump": {
      "href": "http://78.138.66.89:9000/actuator/threaddump",
      "templated": false
    },
    "prometheus": {
      "href": "http://78.138.66.89:9000/actuator/prometheus",
      "templated": false
    },
    "metrics-requiredMetricName": {
      "href": "http://78.138.66.89:9000/actuator/metrics/{requiredMetricName}",
      "templated": true
    },
    "metrics": {
      "href": "http://78.138.66.89:9000/actuator/metrics",
      "templated": false
    }
  }
}
```

Figure 4: Heapdump Actuator Endpoint enabled

```

ptwa@w4deumsy9003863 /cygdrive/w
$ binwalk heapdump.circ | grep 'Private key'
43355707      0x2958E3B      Private key in DER format (PKCS header length: 4, sequence length: 1763)

ptwa@w4deumsy9003863 /cygdrive/w
$ b64 hoSBq52t7mxe910EBYZG1xmrRFQKZBg/hvG26uB0110nn/JtW0PUtPPrI/7At25V66AAQzYpP5yIamqN8IItWdKrtwleTyq7W/STc4BTzwFefm67TdbQJcJ6ULduOd6+5zXjqBpYbhk
5whGsgVoX88eiJ/221/bvgbqJwy1vOS3A00jYiQT2AIoChd0d2bQE274DhwuU6LwVFPpDn7wsFFSEsQVB+Wte6FQ81hMznoEzC2xFzQqXzKuqXTW3hvSAGELyWSBvhbXsT2MplNyz+ykZavuEJv
v14uMbE+9UsUYDCGcggu7o7yZAJUJ8EmYfY18iUbcwR0ZmcM20gveqPwzYbkC4+hmkA2wx0FGKM+Yk10HtkySLUFuSij8jzz1fXxAe12Dy1Ra8ALfpvjHXP0r8VTXjNuhWt9ukYxZD771
mrtVvPZsyv01tqvc161d0C6d86Vw736C17273uVYX00qW2UbUMKKmXI11ieDqcaK8nrjbf0Rz
00000000: 8684 81ab 9dad ee6c 5ef6 5d04 0586 4697 .....I^]....F.
00000100: 19ab 4454 0a64 183f 86f1 b6ea e074 d652 ..DT.d.?.....t.R
00000200: 0e9e 7fc9 b563 8f52 d3cf ac8f fb02 ddb9 ....C.R.....
00000300: 57ae 8001 0cd8 a4fe 7221 a9aa 33c2 08b5 W.....r!..3...
00000400: 60e4 aedc 2579 3caa ed6f d24d ce01 4f3c `...%<.o.M..O<
00000500: 1f79 f9ba ed37 5b40 9723 e949 5db8 e77a .y...7[@.#.I]..z
00000600: fb9c d78e a069 61b8 64e7 0846 b205 685f ....ia.d..F..h_
00000700: cf1e 8a3f f667 5fdb be06 ea27 0cb5 bce4 ...?_g.....
00000800: b703 43a3 608a 93d8 0228 0877 7477 66d0 ..C.'....(wtwf.
00000900: 136e f80e 1c2e 0fa2 f054 53cf 767e f0b1 .n.....TS.v~..
00000a00: f152 12c4 1507 e5ad 7ba1 5006 284c ce7a .R.....{.P.(L.z
00000b00: 04cc 2db1 1734 2a5f 32ae a974 d6de 1bd2 ...-4*_2..t....
00000c00: 0061 0bc9 6481 5616 d7b1 3d8c a4dc b3fb .a..d.V...=....
00000d00: 2919 6afb 8426 fbe5 e2e3 1b13 ef54 b146 .j..&.....T.F
00000e00: 030a 0720 814e e8ef 2640 26e2 49f0 4998 .....N..&@.I.I.
00000f00: 7e26 25f2 251b 7304 4e66 670c db48 2f7a ~&%%.Nfg..H/z
00001000: a3f0 cd86 e40b 8fa1 9e40 36c3 1d1f 18a3 .....e6.....
00001100: 3e62 494e 1ed9 32e4 b505 b928 a3f2 3cf3 >BIN..2....(X<
00001200: d45c 5701 e8b6 943c a245 a040 2dfa 6f8a .\W....<.E.-.o.
00001300: 31d7 3f4a fc55 35e3 ccd8 a15a df6e 918c 1.7J.U5....Z.n.
00001400: 590f bee2 9b0b 5556 966c c825 432c dcea Y.....UV.1.%C...
00001500: bc69 5be1 a3d2 eacd a1e9 5b51 cbbd 9218 .i[.....[Q....
00001600: 0899 cdf9 f551 75d1 f43c 3651 b50c 28a9 .i[.....Qu...<6Q..(
00001700: 955c 8275 89e0 ea71 a281 9d18 db7f 4473 .\u...q.....Ds

ptwa@w4deumsy9003863 /cygdrive/w
$ openssl.exe rsa -in private.key -inform DER -text -noout
RSA Private-Key: (3072 bit, 2 primes)
modulus:
00:86:84:81:ab:9d:ad:ee:6c:5e:f6:5d:04:05:86:
46:97:19:ab:44:54:0a:64:18:3f:86:f1:b6:ea:e0:
74:d6:52:0e:9e:7f:c9:b5:63:8f:52:d3:cf:ac:8f:
fb:02:dd:b9:57:ae:80:01:0c:d8:a4:fe:72:21:a9:
aa:33:c2:08:b5:60:e4:ae:dc:25:79:3c:aa:ed:6f:
d2:4d:ce:01:4f:3c:1f:79:f9:ba:ed:37:5b:40:97:
23:e9:49:5d:b8:e7:7a:fb:9c:d7:8e:a0:69:61:b8:
64:e7:08:46:b2:05:68:5f:cf:1e:8a:3f:f6:67:5f:
db:be:06:ea:27:0c:b5:bc:e4:b7:03:43:a3:60:8a:
93:d8:02:28:08:77:74:77:66:d0:13:6e:f8:0e:1c:
2e:0f:a2:f0:54:53:cf:76:7e:f0:b1:f1:52:12:c4:
15:0f:e5:ad:7b:a1:50:06:28:4c:ce:7a:04:cc:2d:
b1:17:34:2a:5f:32:ae:a9:74:d6:de:1b:d2:00:61:
0b:c9:64:81:56:16:d7:b1:3d:8c:a4:dc:b3:fb:29:
19:6a:fb:84:26:fb:e5:e2:e3:1b:13:ef:54:b1:46:
03:0a:07:20:81:4e:8e:ef:26:40:26:e2:49:f0:49:
98:7e:26:25:f2:25:1b:73:04:4e:66:67:0c:db:48:
2f:7a:a3:f0:cd:86:e4:0b:8f:a1:9e:40:36:c3:1d:
1f:18:a3:3e:62:49:4e:1e:d9:32:e4:b5:05:b9:28:
a3:f2:3c:f3:d4:5c:57:01:e8:b6:94:3c:a2:45:a0:
40:2d:fa:6f:8a:31:d7:3f:4a:fc:55:35:e3:cc:db:
a1:5a:df:6e:91:8c:59:0f:be:e2:9b:0b:55:56:96:
6c:c8:25:43:2c:dc:ea:bc:69:5b:e1:a3:d2:ea:cd:
a1:e9:5b:51:cb:bd:92:18:08:99:cd:f9:f5:51:75:
d1:f4:3c:36:51:b5:0c:28:a9:95:5c:82:75:89:e0:
ea:71:a2:81:9d:18:db:7f:44:73

publicExponent: 65537 (0x10001)
privateExponent:
25:21:f0:d5:09:ea:48:75:9b:e5:30:1a:0b:18:5c:
e6:1b:0c:df:08:bf:f1:8f:8c:01:c9:39:b0:2e:93:
b2:11:e6:bd:3d:13:4f:42:13:6f:4e:ce:f6:18:ac:

```

Figure 5: Private key data found in actuator heapdump

References

- <https://docs.spring.io/spring-boot/docs/current/reference/html/actuator.html#actuator.endpoints>

Severity: High

CVSSv3 Base Score: 7.5 (Vector CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N)

Recommendation

- The Spring Boot Actuator Endpoints should be disabled for release builds, if not absolutely necessary.
- Should they be required, they need to be protected by strong credentials, and if possible restricted to authorized source IP addresses.

5 Appendix

5.1 Description of Severity Levels

The following chapter describes the main features used to assess the severity of vulnerabilities from Chapter 4 of the closure analysis. The explanations of relevant characteristics should form the basis of the assessment process and ensure their traceability. The features summarize security-relevant and technical aspects of a vulnerability.

5.1.1 Severity Level „Informational“

Criteria	Classification / Assessment	Description
Classification	-	The vulnerability poses no direct threat to the application or the software system. An attacker can still use the extracted information to plan and execute further attacks.
Attack Complexity	Not relevant	A consideration of the attack complexity is not relevant for the evaluation.
Impact	Not relevant	Exploitation of the vulnerability does not result in direct damage to the software system or the application.
Likelihood of occurrence	Not relevant	A consideration of the likelihood of occurrence is not relevant for the evaluation.
User interaction	Not relevant	A consideration of the necessity of user interaction is not relevant for the evaluation.
Action required	Not required	A fix is not absolutely necessary. However, it is recommended to fix the cause of the vulnerability in order to avoid an unnecessary security risk.

5.1.2 Severity Level „Low“

Criteria	Classification / Assessment	Description
Classification	-	The vulnerability represents a minor threat to the application or the software system. There is no significant security risk. The combination of multiple vulnerabilities of severity "Low" or the combination with vulnerabilities of a higher severity could still increase the security risk for the application or the software system.
Attack Complexity	High	An attacker usually requires a great deal of (time) effort or considerable - possibly internal - knowledge about the software system or application to successfully exploit the vulnerability.
Impact	Low	By successfully exploiting the vulnerability, an attacker can gain information about the application, cause minor damage to the application, or use the vulnerability to plan and execute further attacks.

		However, the combination with other vulnerabilities can increase the damage effect.
Likelihood of occurrence	Low	Depending on the nature of the vulnerability and its context, successful exploitation may be difficult or only theoretically possible. A low likelihood of occurrence may also be true if no public exploits or descriptions for exploiting the vulnerability are available.
User Interaction	Required if necessary	Depending on the nature of the vulnerability, user interaction may be required. If user interaction is required, significant victim involvement (e.g. social engineering, spam) is often required to provide the attacker with a means of exploiting the vulnerability.
Action Required	Recommended	It is recommended that the cause of the vulnerability be addressed to minimize the security risk. A (productive) use of the software is possible.

5.1.3 Severity Level „Medium“

Criteria	Classification / Assessment	Description
Classification	-	The vulnerability poses an increased security risk to the system or application. An attacker cannot take over the software system or application by exploiting the vulnerability. However, the vulnerability can be used to plan and execute further successful attacks.
Attack Complexity	Moderate to high	An attacker can exploit the vulnerability with significant (time) effort and with targeted knowledge of the software. Exploitation of the vulnerability is usually possible.
Impact	Moderate	An attacker can cause limited direct damage within the application or system and its users by exploiting the vulnerability.
Likelihood of Occurrence	Moderate	Depending on the nature of the vulnerability and its context, successful exploitation is possible. An attack is highly likely to succeed even if no public exploits or descriptions are available to exploit the vulnerability.
User Interaction	May be necessary	Depending on the nature of the vulnerability, unintended user interaction (e.g.: ignoring error messages) may be necessary to provide an opportunity for the attacker to exploit the vulnerability.
Action Required	Recommended	It is strongly recommended to fix the vulnerability. If a fix is not possible (in the short term), the continued use of the software in combination with additional security measures (e.g. monitoring or use of a WAF) is recommended. Since there is an increased security risk, these measures can reduce the probability of occurrence or increase the complexity of the attack.

5.1.4 Severity Level „High“

Criteria	Classification / Assessment	Description
Classification	-	The vulnerability poses a high security risk to the system or application. By exploiting the vulnerability, an attacker can take over parts of the software system, restrict the use of the software or endanger the users and their data of the software system.
Attack Complexity	Low bis moderate	Depending on the nature of the vulnerability and its context, an attacker can exploit the vulnerability with little to reasonable (time) effort and without in-depth knowledge of the software system.
Impact	High	Exploitation of the vulnerability results in high damage within the application or system. By exploiting the vulnerability, an attacker can circumvent security mechanisms (such as the authorization concept or session management) and compromise application data in the system.
Likelihood of Occurrence	High	Depending on the nature of the vulnerability and its context, it is very likely that the vulnerability will be exploited with a low attack complexity. The probability of a successful attack increases if the vulnerability is known and (at least) one public exploit is available.
User interaction	Possibly necessary	Active user interaction is usually not required. If user interaction is necessary, it requires increased attention from the user to detect an attack (for example, clicking a malicious link).
Action Required	Required	The cause of the weaknesses must be eliminated quickly. The system or application must not be used (productively) because there is a high security risk for the operation and the user.

5.1.5 Severity Level „Critical“

Criteria	Classification / Assessment	Description
Classification	-	The vulnerability poses a very high security risk to the system or application. Exploitation of the vulnerability allows an attacker to take over the entire system or IT infrastructure.
Attack Complexity	Low	Depending on the type of vulnerability and its context, a successful attack usually requires little effort or medium to little technical knowledge.

Impact	Critical	An attacker is able to compromise the system or the IT infrastructure and execute malicious actions on operating system level or other systems of the IT infrastructure.
Likelihood of Occurrence	High to very high	Due to the low complexity of the attack there is a high probability of occurrence. The probability increases if the vulnerability is known and a public exploit is available.
User Interaction	Not required	User interaction is not required in most cases.
Action Required	Immediately required	The vulnerability must be fixed as soon as possible. The system must not be used (productively) under any circumstances, as there is a very high security risk for the IT infrastructure, the users of the system and thus for business operations.