Report External Penetration Test

Thinkwise Platform

Version 1.2, May 8th 2023

Date	Version	Comments
27/02/2023	0.1	Initial draft for internal review
22/03/2023	1.0	Final version for client
03/05/2023	1.1	Updated version after retest
08/05/2023	1.2	Updated after customer feedback

Table of Contents

1 – Management Summary

Thinkwise B.V. (from hereon: Thinkwise) is creator and owner of the Thinkwise Low Code software platform, which can be used to build robust software applications for clients in a short period of time, making use of the functionality and technology that is part of the platform.

To be able to prove to customers that the Thinkwise Low Code platform is sufficiently secure, Thinkwise has mandated nSEC/Resilience to perform a penetration test on the platform. This penetration test was expanded with a number of audit-like checks.

During the initial penetration test on the example application of the Thinkwise platform no findings of high or critical severity were done. Testers did not succeed in obtaining significant amounts of sensitive data nor take control of the server. This is a good result.

The audit also did not produce findings that required immediate actions.

After receiving feedback from Thinkwise on the initial findings, and retesting of fixed findings, only a small number of low severity findings remained. These findings of low severity can be seen as findings for which there is no direct urgency to address them, but implementing solutions for these findings will further raise the bar for potential attackers and establish a baseline that would give customers or external auditors increased confidence.

Chapter 6 of the report provides a complete overview of all the findings. Below is an overview giving the results, per category (as identified by OWASP).

Topic/area	Test result
Network level	Good – no vulnerabilities found
Auth & session management	Findings –2x low severity
Broken Access Control	Good – no vulnerabilities found
Unrestricted File Upload	Good – no vulnerabilities found
Directory traversal / File inclusion	Good – no vulnerabilities found
Cross-site scripting (XSS)	Good – no vulnerabilities found
SSL/TLS	Good – no vulnerabilities found
SQL injection	Good – no vulnerabilities found
Error handling	Good – no vulnerabilities found
Sensitive data exposure	Finding – 1x low severity
Security (mis)configuration	Good – no vulnerabilities found
Audit	Good – no vulnerabilities found

2 – Scope and context

Thinkwise B.V. (from hereon: Thinkwise) is creator and owner of the Thinkwise Low Code software platform, which can be used to build robust software applications for clients in a short period of time, making use of the functionality and technology that is part of the platform.

To be able to prove to customers that the Thinkwise Low Code platform is sufficiently secure, Thinkwise has mandated nSEC/Resilience to perform a penetration test on an example application built using the platform.

Due to the nature of the application / platform, nSEC/Resilience has advised to also consider adding audit elements to the security evaluation on the platform. These audit elements will expand the coverage of the test activities to include more elements that are difficult to cover from an external dynamic penetration test. The results of this white box audit are also included in this report.

The attack surface (areas of the information system that an attacker or security evaluator can choose to initiate an attack) for the penetration test was defined as, and limited to:

- https://nsecresilience.thinklab.cloud/ (General)
- https://nsecresilience-prod.thinklab.cloud/universal/ (User interface)
- https://nsecresilience-prod.thinklab.cloud/indicium/iam/insights (Application tier)

It was explicitly allowed as part of the penetration test to investigate and exploit vulnerabilities in the assets in scope, as long as direct attack surface was limited to the definition above.

During the penetration test forensic research, code reviews and exhaustive DDOS testing were out of scope.

3 – Test approach and process evaluation

For the penetration test, 40 hours of testing was allocated. The testers received three test accounts for the Insights application, each with different access rights, so that proper tests for access control could take place. As such, the penetration test was executed grey-box.

Methodologically, the penetration test was performed in line with the PTES (infrastructure level) and OWASP WSTG (application level).

Reconnaissance for the penetration test was performed with industry-standard tooling (scanners and scripts) and by manually searching through public available sources. At network level also open ports and active services were investigated.

During the execution- and exploitation phase various tools were used. However, majority of the checks were performed manually, where internet traffic was investigated and manipulated with proxy tooling.

For the audit elements two resource days were allocated. The activities were for the largest part performed on location in Apeldoorn by two resources working in parallel.

No issues occurred during test execution.

After initial testing, Thinkwise gave feedback on the communicated findings. Some findings were not regarded as risk (or regarded as working as designed) and one finding was fixed and retested. This report describes the results after retest.

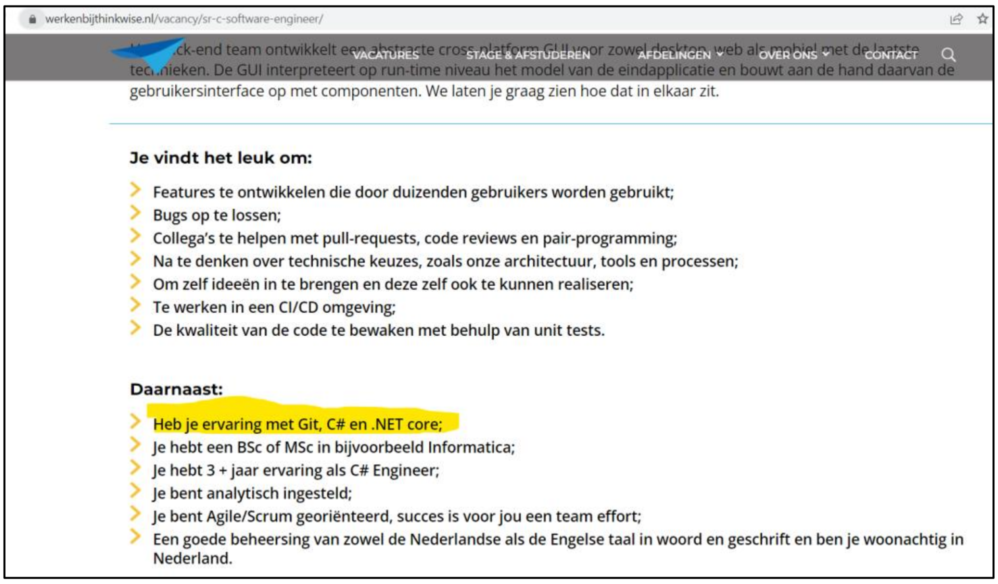
4 – Reconnaissance phase

Goal of the reconnaissance phase is to collect data from public sources and non-intrusive scans. Results from the reconnaissance phase give valuable information to be used in the execution phase.

4.1 – Public information sources

The following results are gathered from the open source intelligence reconnaissance phase (only the most relevant results are mentioned):

- See the full list of subdomain enumeration in Appendix A – DNS reconnaissance
- The identified domains have been checked using a tool that automatically makes screenprints based on a list of domains. No relevant results followed from these checks
- The IP addresses that corresponded to the identified subdomains were added to the scope for the external network level checks
- Search engine reconnaissance showed that job vacancy information contains technology information:



...ck-end team ontwikkelt een abstracte cross-platform GUI voor zowel desktop, web als mobiel met de laatste tec..nieken. De GUI interpreteert op run-time niveau het model van de eindapplicatie en bouwt aan de hand daarvan de gebruikersinterface op met componenten. We laten je graag zien hoe dat in elkaar zit.

Je vindt het leuk om:

- Features te ontwikkelen die door duizenden gebruikers worden gebruikt;
- Bugs op te lossen;
- Collega's te helpen met pull-requests, code reviews en pair-programming;
- Na te denken over technische keuzes, zoals onze architectuur, tools en processen;
- Om zelf ideeën in te brengen en deze zelf ook te kunnen realiseren;
- Te werken in een CI/CD omgeving;
- De kwaliteit van de code te bewaken met behulp van unit tests.

Daarnaast:

- Heb je ervaring met Git, C# en .NET core;
- Je hebt een BSc of MSc in bijvoorbeeld Informatica;
- Je hebt 3 + jaar ervaring als C# Engineer;
- Je bent analytisch ingesteld;
- Je bent Agile/Scrum georiënteerd, succes is voor jou een team effort;
- Een goede beheersing van zowel de Nederlandse als de Engelse taal in woord en geschrift en ben je woonachtig in Nederland.

- Search for externally exposed software repositories did not yield results
- No other results were obtained from open source reconnaissance

4.2 – Vulnerability scan and manual reconnaissance

In addition to the collection of data from public sources, several vulnerability scans were performed with various tools, including the industry standard tool NetSparker and OWASP ZAP. Additionally, manual reconnaissance was performed to identify the application stack and attack vectors for the execution phase. The most relevant results:

- Manual reconnaissance reports that the target application is built on a Microsoft stack (IIS/10.0) with ASP.net
- Brute force directory and file scanners produced only a few results, none of which were relevant for the further investigation
- Exploration of the workflow shows various functionality that can be explored further, in particular upload functionality and preview functionality
- Many of the notifications from the scans were also found with manual checks/tests and are described in the paragraphs of Chapter 5

4.3 – Network level scan

In addition to the analysis of public sources, various scans have been performed on network level. These scans were performed using at least the industry standard tools Nessus and Nmap.

The main IP address (20.170.5.72) was scanned on network level, with the following results:

Port	Protocol	Status	Service	Details
80	TCP	Open	HTTP	Azure application gateway
443	TCP	Open	HTTPS	Azure application gateway

Next to the regular TCP network level scans, various firewall evasion techniques were deployed and UDP scanning was executed. From these checks, no findings were done on network level for the main IP address in scope.

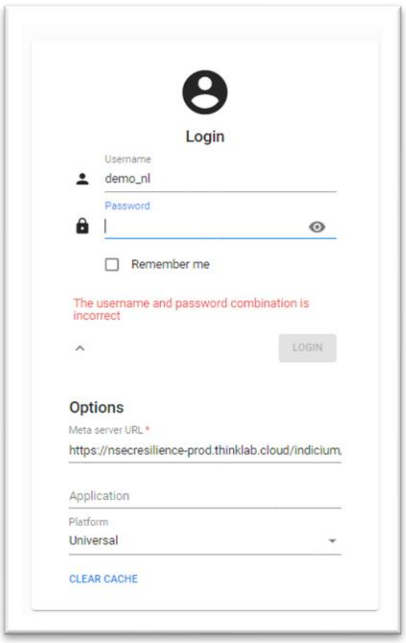
5 – Penetration Test: application level

Following the outcomes of the reconnaissance phase, a number of aspects are analyzed in more detail. The executed actions and analyses are discussed in the following paragraphs, classified according the OWASP top 10.

5. 1 – Authentication and Session management

5.1.1 – Login mechanism

Upon login to the insights application (user interface) via https://nsecresilience-prod.thinklab.cloud/universal/, by default a username and password was required before any other interaction with the application was possible:


Login

Username: demo_nl

Password:

Remember me

The username and password combination is incorrect

LOGIN

Options

Meta server URL *: https://nsecresilience-prod.thinklab.cloud/indicium

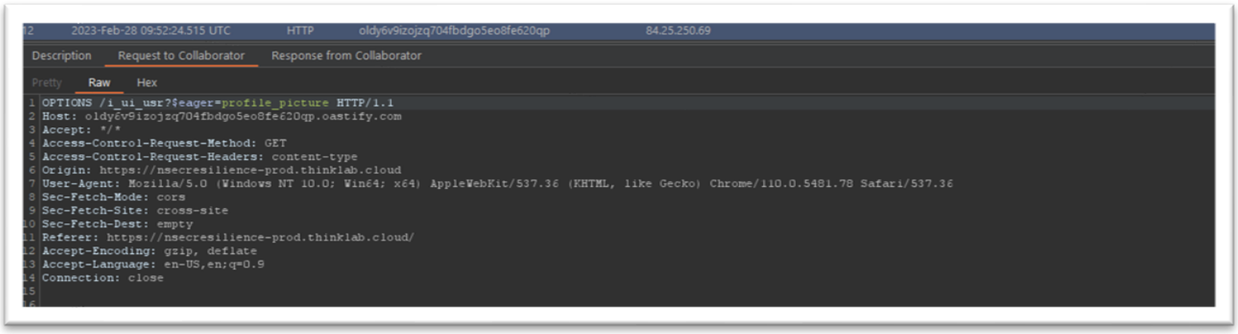
Application

Platform: Universal

CLEAR CACHE

When either the username or password is incorrect, a neutral error message is returned. This prevents the login mechanism from being used for user enumeration which is good. On the login screen, a meta server URL and an application and platform could be configured.

It was attempted to change the meta server URL to a server in control of the testers, to see if credentials could be intercepted in this way. Doing so it was noticed that one request was intercepted by the server in our control:

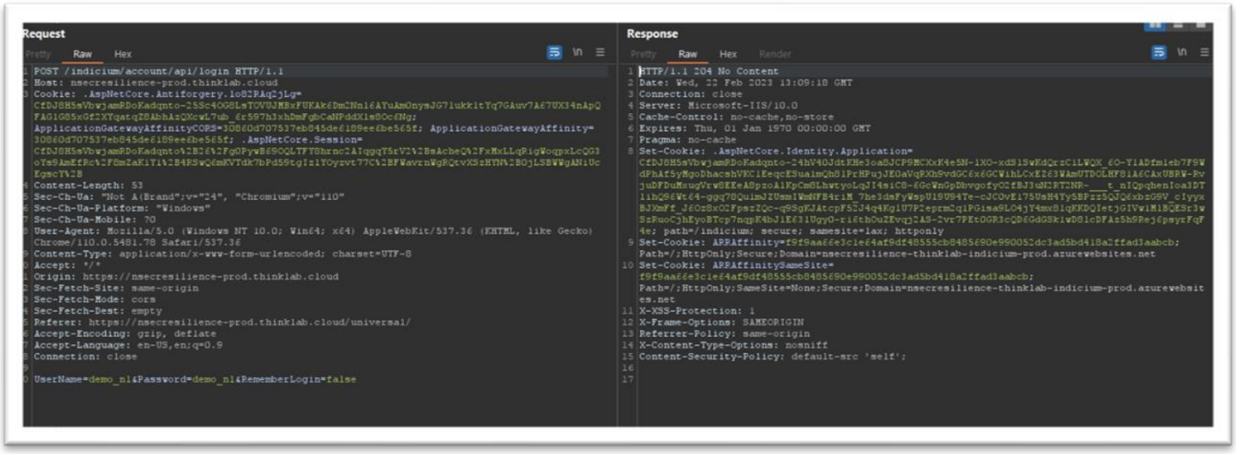


```
12    2023-Feb-28 09:52:24.515 UTC    HTTP    oldy6v9izojzq704fbdgo5eo8fe620qp    84.25.250.69
```

```
OPTIONS /i_ui_usr?$eager=profile_picture HTTP/1.1
Host: oldy6v9izojzq704fbdgo5eo8fe620qp.oastify.com
Accept: */*
Access-Control-Request-Method: GET
Access-Control-Request-Headers: content-type
Origin: https://nsecresilience-prod.thinklab.cloud
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36
Sec-Fetch-Mode: cors
Sec-Fetch-Site: cross-site
Sec-Fetch-Dest: empty
Referer: https://nsecresilience-prod.thinklab.cloud/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

```

The source IP address however was an attacker IP address and not an IP address controlled by Thinkwise (meaning that no SSRF attacks were possible).

When submitting the login credentials a POST request is made to /indicium/account/apio/login with the username and password in the request body:



Request

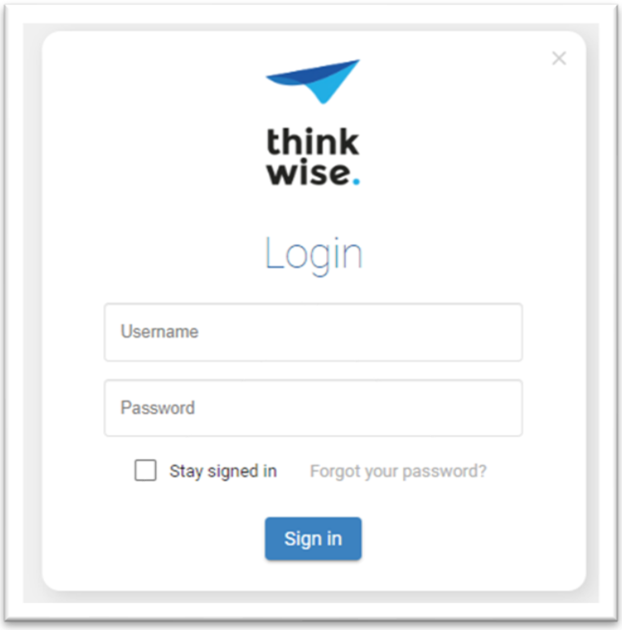
```
POST /indicium/account/api/login HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Antiforgery.lo82RAq2jLg=
CfDJ8H5sVbwjamRDoKadqnto-25Sc4OG8LsTOVUJMBxFUKAk6Dm2Nnl6AYuAmOnysJG7lukkltYq7GAuv7A67UX34nApQ
FAG1G85xGf2XYqatqZ8AbhAzQXcwL7ub_6r597h3xhDmFgbCaNPddXlm8Oc6Ng;
ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be565f; ApplicationGatewayAffinity=
30860d707537eb845de6189ee6be565f; .AspNetCore.Session=
CfDJ8H5sVbwjamRDoKadqnto%2B26%2FgOPywB69OQLTFY8hrnc2AIqgqY5rV2%2BsAcheQ%2FxMxLLqRigWoqpxLcQG1
oYs9AmEfRc%2F8mZaKiYi%2B4RSwQ6mKVTdk7bPd59tgIz1YOyzvt77C%2BFWavrnWgRQtvXSzHYN%2BOjLSBWWgANiUc
EgscY%2B
Content-Length: 53
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Sec-Ch-Ua-Platform: "Windows"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept: */*
Origin: https://nsecresilience-prod.thinklab.cloud
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://nsecresilience-prod.thinklab.cloud/universal/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

UserName=demo_nl&Password=demo_nl&RememberLogin=false
```

Response

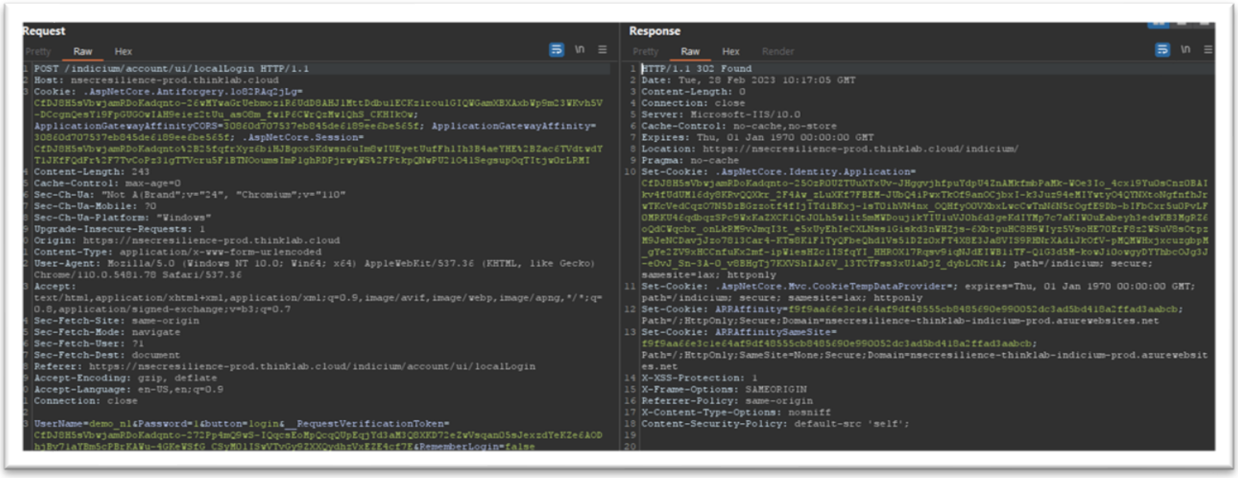
```
HTTP/1.1 204 No Content
Date: Wed, 22 Feb 2023 13:09:18 GMT
Connection: close
Server: Microsoft-IIS/10.0
Cache-Control: no-cache,no-store
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Pragma: no-cache
Set-Cookie: .AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-24hV40JdtKHe3oa8JCP9MCXxK4e5N-1XO-xdS1SwKdQrzC1LWQX_6O-Y1ADfmleb7F9W
dPhAf5yMgoDhacshVKC1EeqcESua1mQh81PrHPujJE0aVqRXh9vdGC6x6GCWihLCxEZ63WAmUTDOLHF81A6CAxUBRW-Rv
juDFDuMxugVrw8EEeA8pzoA1KpCm8LhwtyoLqJI4siC8-6GcWnGpDbvgofyO2fBJ3uN2RT2NR-___t_nIQpqhenIoa3DT
1ihQ96Wt64-ggq78QuimJ2UsmIWmNFB4riM_7he3dsFyWspU19U94Te-cJCOvE175UsH4Yy5BPzz5QJQ6xbzG9V_c1yyx
BJXmFf_J6Oz8xO2FpszZQc-q9SgKJAtcpF52J4q4Kg1U7P2eprm2q1PGisa9LO4jY4mx81qKKDQIetjGIVw1M1BQESr3w
SzRuoCjhEyoBTcp7nqpK4bJ1E631UgyO-ri6thOuZEvqj2AS-2vr7PEtOGR3cQD6GdGSk1wD81cDFAz5h9Rej6psyrFqF
4e; path=/indicium; secure; samesite=lax; httponly
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=
f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsit
es.net
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self';

```

A 204 “No Content” response is then returned together with an *.AspNetCore.Identity.Application* cookie, which acts as session cookie for the application. When successfully authenticated, this cookie is used for authorizing further requests made in the application.

An alternative login function for indicium local login was found via https://nsecresilience-prod.thinklab.cloud/indicium/account/ui/localLogin:



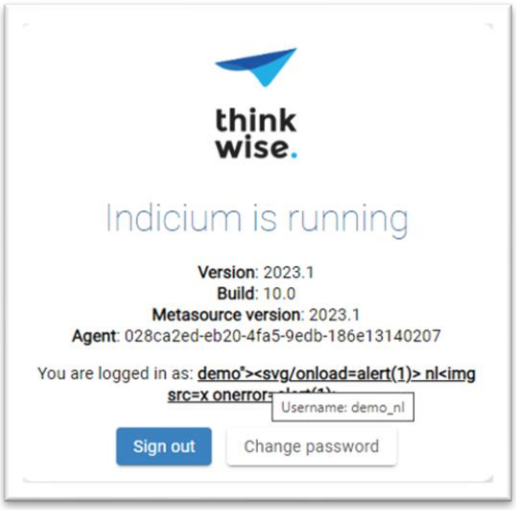
Similar to the previous login, upon successful authentication a *.AspNetCore.Identity.Application* is set.



```
POST /indicium/account/ui/localLogin HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Antiforgery.1o82RAq2jLg=
CfDJ8H5sVbwjamRDoKadqnto-26wMYwaGrUebmoziR6UdD8AHJ1MttDdbu1ECKz1rou1GIQWGamXBXAxbWp9m23WKvh5V
-DCcgnQesYi9FpGUGOwIAH9eiez2tUu_asO8m_fw1P6CWrQzMwlQhS_CKHIkOw;
ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be565f; ApplicationGatewayAffinity=
30860d707537eb845de6189ee6be565f; .AspNetCore.Session=
CfDJ8H5sVbwjamRDoKadqnto%2B25fqfrXyz6biHJBgoxSKdwsn6uIm8wIUEyetUufFh1Ih3B4aeYHE%2BZac6TVdtwdY
T1JKfFQdFr%2F7TvCoPz31gTTVcru5F1BTNOoumsImP1ghRDPjrwyWS%2FPtkpQNwPU21O41SegsupOqTItjwOrLRMI
Content-Length: 243
Cache-Control: max-age=0
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://nsecresilience-prod.thinklab.cloud
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
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=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://nsecresilience-prod.thinklab.cloud/indicium/account/ui/localLogin
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

UserName=demo_nl&Password=1&button=login&__RequestVerificationToken=
CfDJ8H5sVbwjamRDoKadqnto-272Pp4mQ9wS-IQqcsEoMpQcqQUpEqjYd3aM3Q8XKD72eZwVsqan05sJexzdYeKZe6AOD
hjBv71aYBm5cPBrKAWu-4GKeWSfG_CSyM01ISwVTvGy9ZXXQvdhzVxEZE4cf7E&RememberLogin=false
```

```
HTTP/1.1 302 Found
Date: Tue, 28 Feb 2023 10:17:05 GMT
Content-Length: 0
Connection: close
Server: Microsoft-IIS/10.0
Cache-Control: no-cache,no-store
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Location: https://nsecresilience-prod.thinklab.cloud/indicium/
Pragma: no-cache
Set-Cookie: .AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-25OzROUZTUuXYxUv-JHggvjhfpuYdpU4ZnAMkfmbPaMk-WOe3Io_4cx19YuOsCnzOBAI
kv4fUdUMl6dy8KRvQQXkr_2F4Aw_zLuXKf7FBEM-JUbQ4iPwxTkOf9anOCjbxI-k3Juz94eMIYwtyO4QYNXtoNgfnfhJr
wTKcVedCqzO7N5DzBGzzotf4fIjITdiBKxj-1sTOihVN4nx_OQHfyOOVXbxLwcCwTnN6N5rOgfE9Db-bIFbCxr5uOPvLF
OMRKU46qdbqzSPc9WxKaZXCK1QtJOLh5w1lt5mMWDoujikYIU1uVJOh6d3geKdIYMp7c7aKIWOuEabeyh3edwKB3MgRZ6
oQdCWqcbr_onLkRM9vJmqI3t_e5xUyEhIeCXLNss1Giskd3nWHZjs-6XbtpuHC8H9WIyz5VsoHE70ErF8z2WSuV8sOtpz
M9JeNCDavjJzo7813Car4-KTs8KiF1TyQFbeQhd1Vs51DZzOxFT4X8E3Ja8VIS9RHNrXAdiJkOfV-pMQMWHxjxcuzgbpM
_gYe2ZV9xHCCnfuKx2mf-ipWiesHZc1ISfqYI_HHROX17Rqsv9iqNJdEIWB1iTF-Q1G3d5M-kowJiOowgyDYYhbcOJg3J
-eOvJ_Sn-3A-O_v8BHgTj7KXVShIAJ6V_13TCYFss3xUlaDjZ_dybLCNtiA; path=/indicium; secure;
samesite=lax; httponly
Set-Cookie: .AspNetCore.Mvc.CookieTempDataProvider=; expires=Thu, 01 Jan 1970 00:00:00 GMT;
path=/indicium; secure; samesite=lax; httponly
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=
f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsit
es.net
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self';
```

This cookie holds the same authorization as the cookie that is returned by the user interface login however upon successful login the user is not immediately redirected to the user interface.



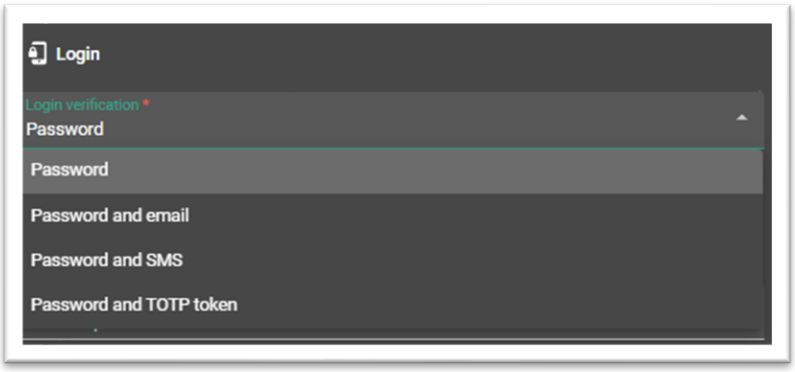
Indicium is running

Version: 2023.1
Build: 10.0
Metasource version: 2023.1
Agent: 028ca2ed-eb20-4fa5-9edb-186e13140207

You are logged in as: demo"><svg/onload=alert(1)> nl<img src=x onerror=alert(1);

Username: demo_nl

Using a higher privileged admin account with access to IAM, further login options could be configured, directly affecting the login mechanism.



Login verification *: Password

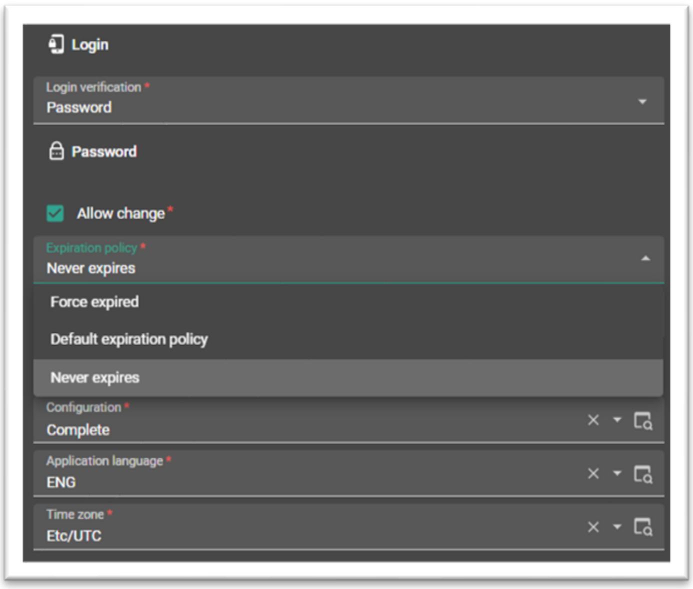
In the login mechanisms as configured, no vulnerabilities were identified.

5.1.2 – Password change mechanism

For the Insight application there are possibilities to change user passwords:

- By regular users, for their own accounts (if enabled in the IAM)
- By a higher privileged user

The highly privileged user is responsible for setting appropriate security configurations such as password strength and expiration policy. Settings such as password expiry are not set by default.


Login

Login verification *
Password

Password

✓ Allow change *

Expiration policy *
Never expires

Force expired
Default expiration policy
Never expires

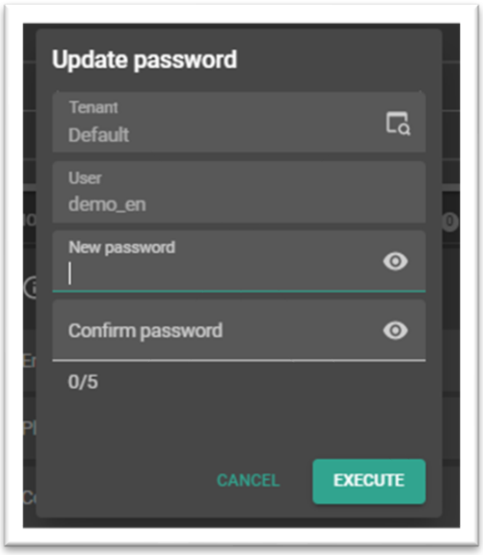
Configuration *
Complete

Application language *
ENG

Time zone *
Etc/UTC

Password change for IAM user (high privilege):

The IAM user (high privilege) is able to change passwords on behalf of users via the "update password" function. The password policy can be configured via IAM.


Update password

Tenant
Default

User
demo_en

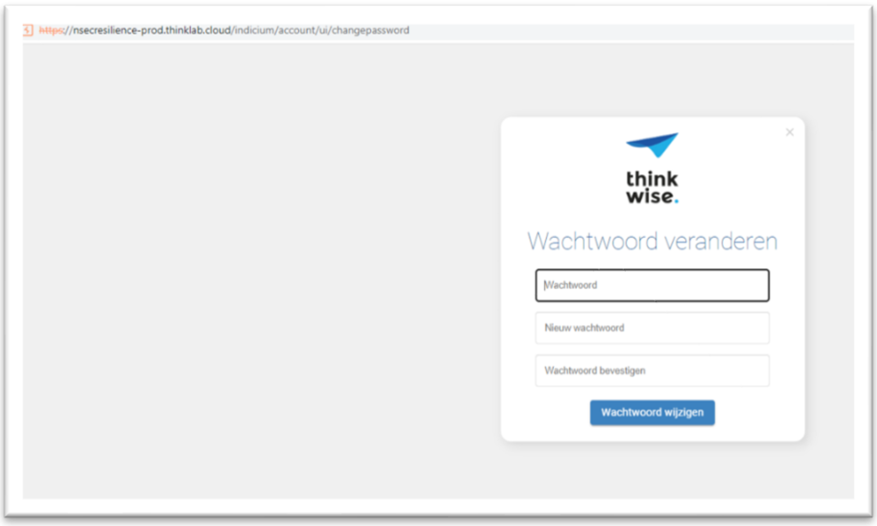
New password

Confirm password

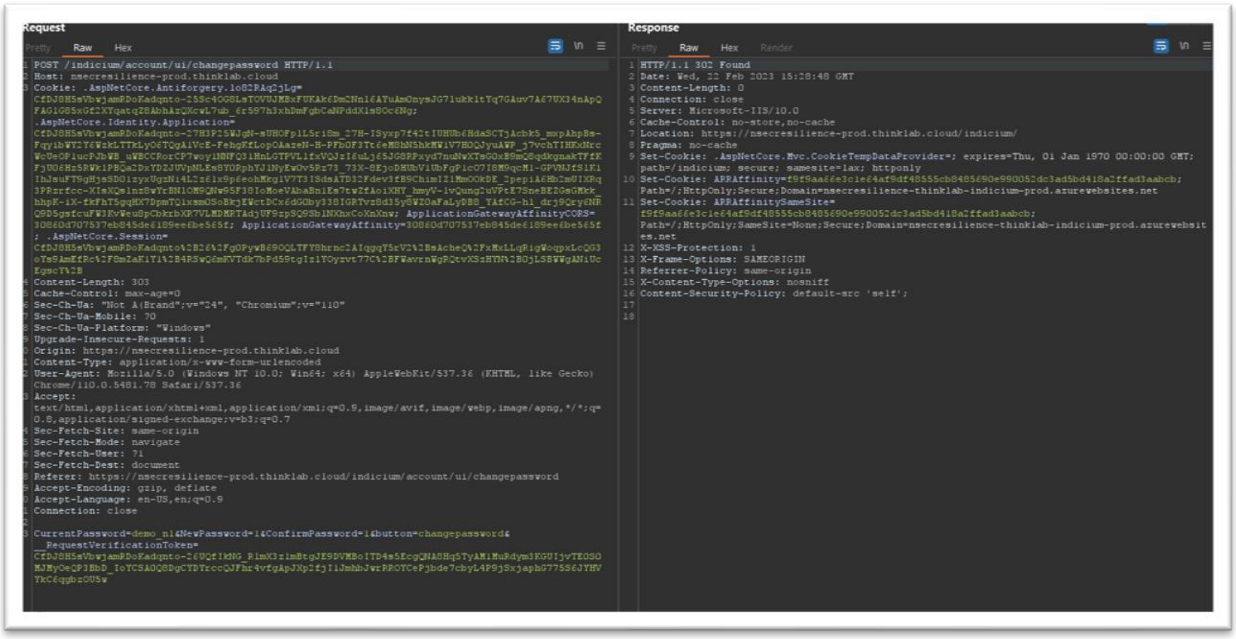
0/5

Password change for Insights user (low privileged):

If enabled by the IAM user, a lower privileged insights application user is able to change their own password. In this case the current password is required, which is good:


https://nsecresilience-prod.thinklab.cloud/indicium/account/ui/changepassword

In the backend this current password is also passed on and a valid session cookie is required for the request to be successful:



```
POST /indicium/account/ui/changepassword HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Antiforgery.lo82RAq2jLg=
CfDJ8H5sVbwjamRDoKadqnto-25Sc4OG8LsTOVUJMBxFUKAk6Dm2Nnl6AYuAmOnysJG71ukk1tYq7GAuv7A67UX34nApQ
FAG1G85xGf2XYqatqZ8AbhAzQXcwL7ub_6r597h3xhDmFgbCaNPddXls8Oc6Ng;
.AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-27H3P25WJgN-sUHOFplL5ri8m_27H-ISyxp7f42tIUHUb6HdaSCTjAcbk5_mxpAhpBs-
FqyibWY2Y6WzkLTTkLyO6TQgA1VcE-FehgKfLopOAazeN-H-PFbOF3Tt6eM8hN5hkMW1V7HOQJyuAWP_j7vchTIHKxNrc
WcUeOPlucPJbWB_uWBCCRorCP7woyiNNFQ31HnLGTPVL1fxVQJzI6uLj65JG8PPxyd7nuNwXTsGOxB9mQ8qdkgnakTFfK
FjUO6Hz5RWklPBQa2DxYDZJUVpNLEs8YORphYJ1NyEwOv5Rz73_73X-8EjoDHUbV1UbFgP1cO7I8M9qcMl-GPVNJfS1K1
IhJsuFT9gHjsSDOlzyxUgzNi4L2z61x9p6eohMkglV7T3ISdsATD32Fdev3fB9ChimI21MmOOkDE_2pepiA6Hb2mUIXPq
3PPzrfcc-XIsXQslnz8wYrBN1OM9QNw95F38IoMoeVAbaBn1Es7twZfAoiXHY_hmyV-1vQung2uVPtE7SneBE2GsGMkk_
hhpK-iX-fkFhT5gqHX7DpmTQ1xsmOSoBkjEWctDCx6dGGby338IGRTvz8d35y8WZOaFaLyDB8_YAfCG-hl_drj9Qry6NR
Q9D5gsfcuFW3KvWeu8pCbkrbXR7VLMDMRTAdjUF9zpSQ9SblNXhxCoXnXnw; ApplicationGatewayAffinityCORS=
30860d707537eb845de6189ee6be565f; ApplicationGatewayAffinity=30860d707537eb845de6189ee6be565f
; .AspNetCore.Session=
CfDJ8H5sVbwjamRDoKadqnto%2B26%2FgOPywB69OQLTFY8hrnc2AIqqqY5rV2%2BsAcheQ%2FxMxLLqRigWoqpxLcQG3
oYs9AmEfRc%2F8mZaK1Y1%2B4RSwQ6mKVTdk7bPd59tgIz1YOyzvt77C%2BFWavrnWgRQtvXSzHYN%2BOjLSBWWgANiUc
EgscY%2B
Content-Length: 303
Cache-Control: max-age=0
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://nsecresilience-prod.thinklab.cloud
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
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=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://nsecresilience-prod.thinklab.cloud/indicium/account/ui/changepassword
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

CurrentPassword=demo_nl&NewPassword=1&ConfirmPassword=1&button=changepassword&
__RequestVerificationToken=
CfDJ8H5sVbwjamRDoKadqnto-26UQfIkNG_PlmX3zlmBtgJE9DVMBoITD4s5EcgQNA8Hq5TyAMiMuRdym3KGUIjvTEOSO
MJMyOeQP3BbD_IoYCSAOQ8DgCYDYrccQJFhr4vfgApJXp2fjIlJmhbJwrRPOYCePjbde7cbyL4P9jSxjaphG775S6JYHV
YkC6qgbzOU5w
```

```
HTTP/1.1 302 Found
Date: Wed, 22 Feb 2023 15:28:48 GMT
Content-Length: 0
Connection: close
Server: Microsoft-IIS/10.0
Cache-Control: no-store,no-cache
Location: https://nsecresilience-prod.thinklab.cloud/indicium/
Pragma: no-cache
Set-Cookie: .AspNetCore.Mvc.CookieTempDataProvider=; expires=Thu, 01 Jan 1970 00:00:00 GMT;
path=/indicium; secure; samesite=lax; httponly
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=
f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsit
es.net
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self';
```

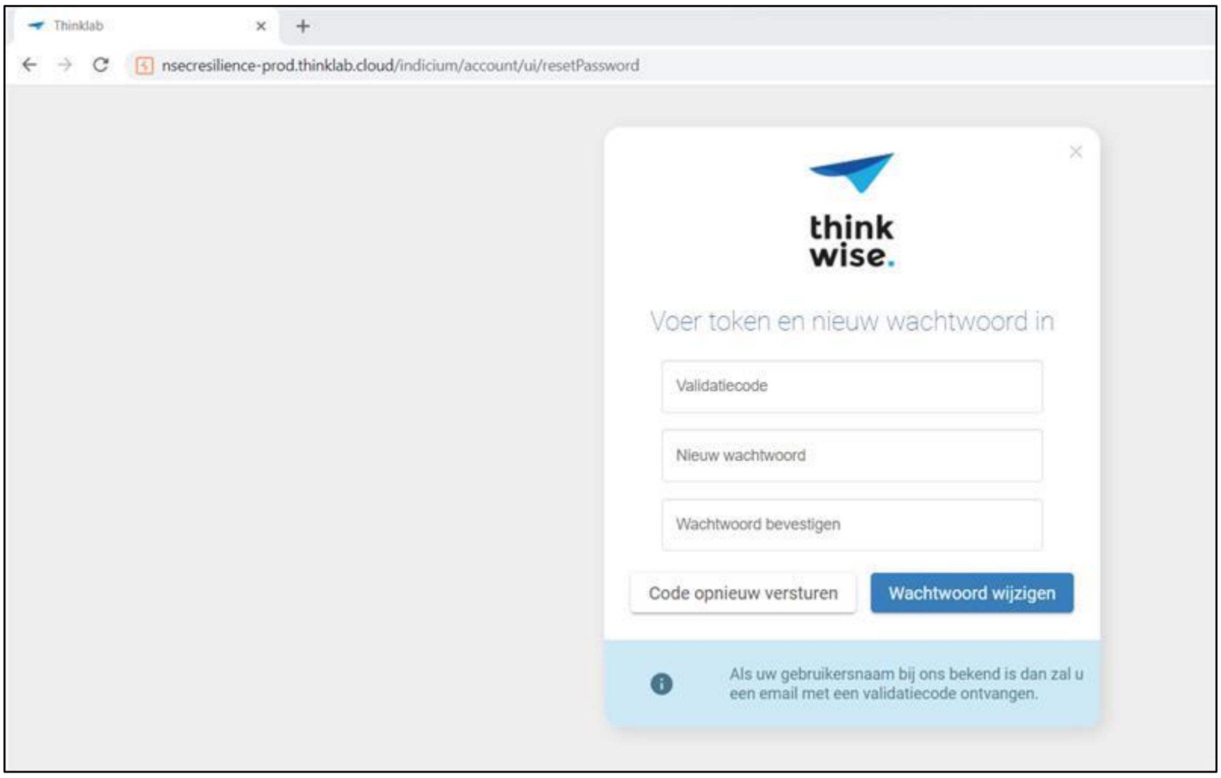
It is in general recommended to enforce password policies by default instead of making it optional for the high privileged user to configure. This will help push the use of stronger passwords for applications and users making use of the Thinkwise software. However because the password policy can be defined by any organization through IAM no finding is included.

Password reset (unauthenticated)

The Insights application also contained a password reset mechanism in the form of a "password forgot" function for unauthenticated users (if password recovery is allowed by the administrator).

In this case a username must be entered, after which an email with a unique token is sent via e-mail that can be used to reset the password.

This function is not sensitive to username enumeration; the same response is given regardless of whether the input was correct or not.


Thinklab
nsecresilience-prod.thinklab.cloud/indicium/account/ui/resetPassword
think wise.

Voer token en nieuw wachtwoord in

Validatiecode

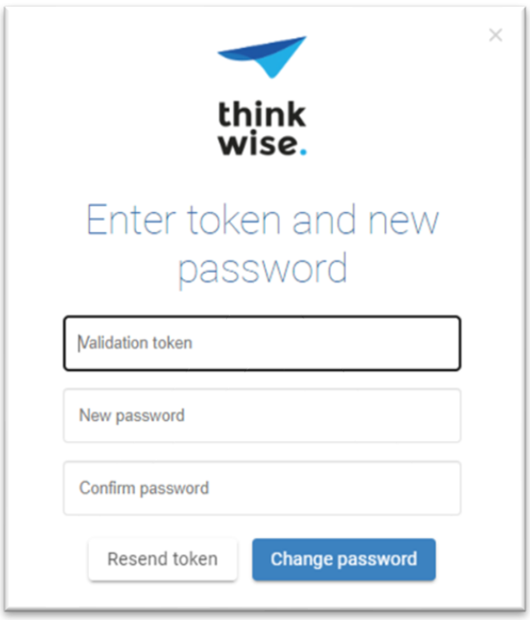
Nieuw wachtwoord

Wachtwoord bevestigen

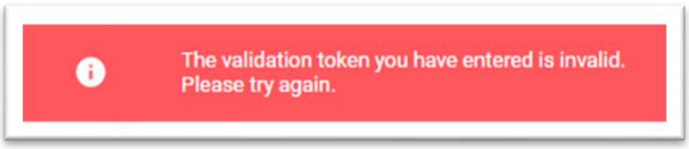
Code opnieuw versturen
Wachtwoord wijzigen

Als uw gebruikersnaam bij ons bekend is dan zal u een email met een validatiecode ontvangen.

After an existing username is entered, the user should receive an email with a validation token that has to be entered together with a new password:

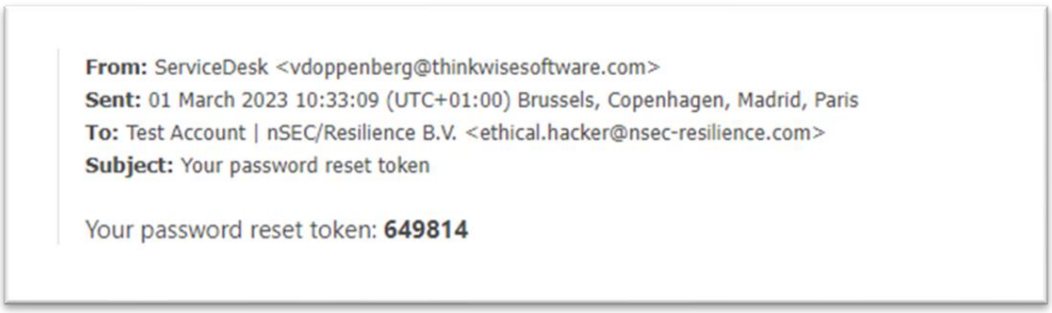


When an invalid or no validation token is entered, an error is returned indicating that the token is incorrect:



The validation token you have entered is invalid.
Please try again.

When checking the received e-mail it is observed that the validation token is a 6-digit code, for example:



From: ServiceDesk <vdoppenberg@thinkwisesoftware.com>
Sent: 01 March 2023 10:33:09 (UTC+01:00) Brussels, Copenhagen, Madrid, Paris
To: Test Account | nSEC/Resilience B.V. <ethical.hacker@nsec-resilience.com>
Subject: Your password reset token

Your password reset token: **649814**

This code is tested to have an expiration time of under 30 minutes, which makes brute force attacks on this code less likely. However there is no brute force attack detection present on the MFA function. An attacker seems to be able to try many combinations. It is advised to add more protection on the backend by for example temporarily blocking a user when more than 5 invalid MFA codes are submitted in a short timeframe. A finding of severity low was added for this.

5.1.3 – Session management

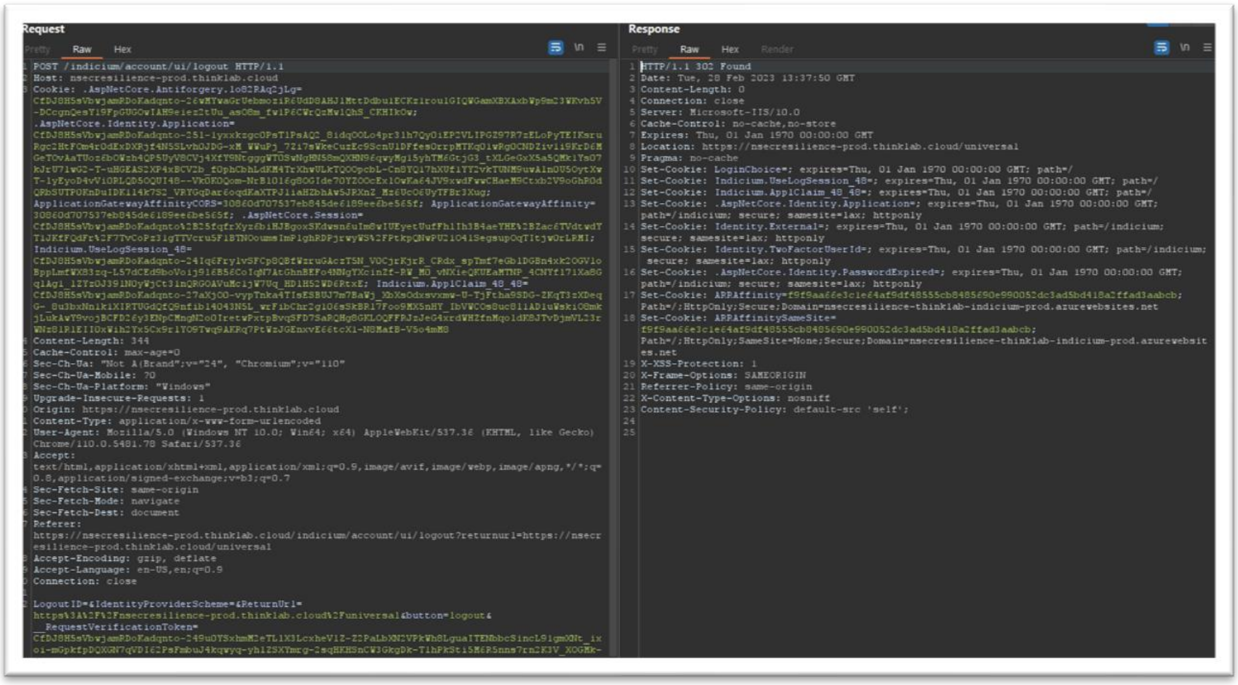
Checks were done for session management such as checks around validity, lifetime and data storage.

It was found that sessions do not end within a specific period of inactivity or after browser close. In some cases sessions remain active for longer periods due to specific requests generated from the GUI keeping the session alive. The session is ended on browser close (unless the user has selected the “remember me” option).

Generally it is desired from the perspective of security to by default have a mechanism for automatic session timeout after a short period of inactivity (a few hours maximum) that works in all situations, so that the user is forced to reauthenticate after not using an application for some time.

A finding of severity low was added for this observation, with the recommendation to enforce that a session timeout is implemented by default and is active in all scenario’s.

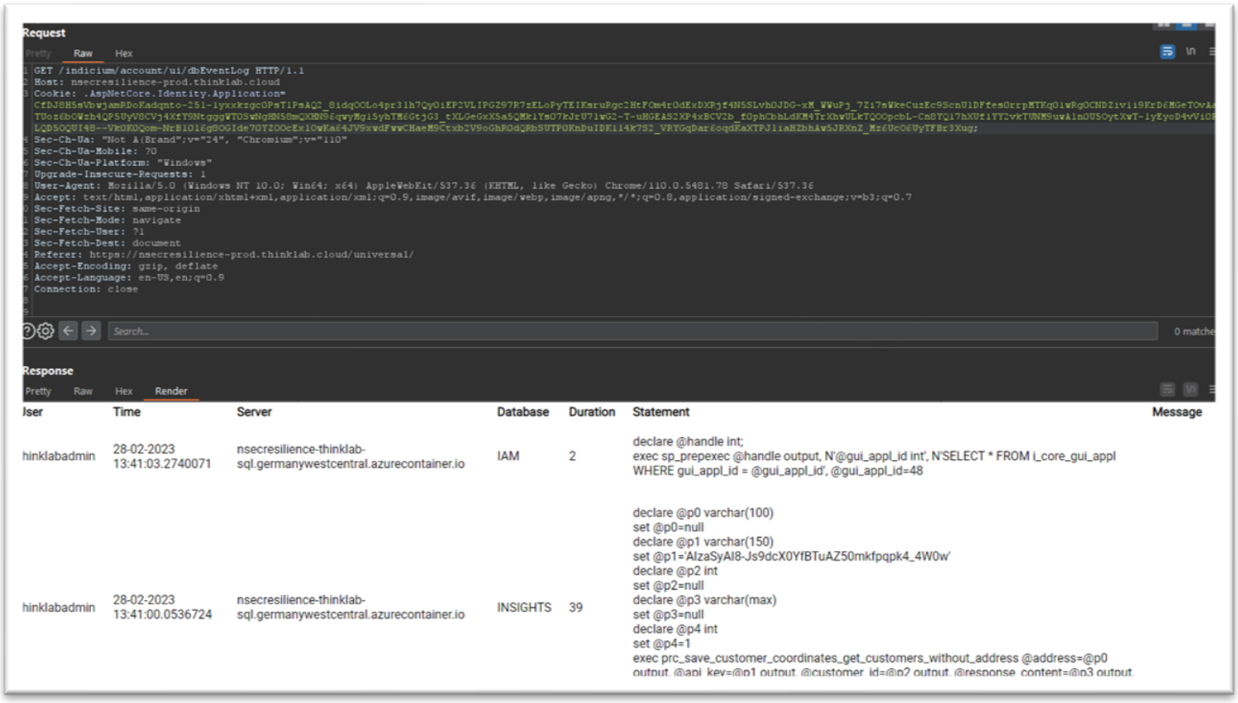
It was also found that after a user explicitly logs out via the logout function, requests made with the old session cookie (*.AspNetCore.Identity.Application*) would still return a valid response, some time after the user had logged out.



```
POST /indicium/account/ui/logout HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Antiforgery.1o82RAq2jLg=
CfDJ8H5sVbwjamRDoKadqnto-26wMYwaGrUebmoziR6UdD8AHJ1MttDdbuIECKz1rouIGIQWGamXBXAxbWp9m23WKvh5V
-DCcgnQesYi9FpGUGOwIAH9eiez2tUu_as08m_fw1P6CWrQzMw1QhS_CKHIkOw;
.AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-251-1yxxkzgcOPsT1PsAQC_8idqOOLo4pr31h7QyOiEP2VLIPGZ97R7zELoPyTEIKsru
Rgc2HtFOm4rOdExDXRjf4N5SLvhOJDG-xM_WWuPj_72i7sWkeCuzEc9ScnU1DFfesOrrpMTKqO1wRgOCNDZivii9KrD6M
GeTOvAaTUoz6bOWzh4QP5UyV8CVj4XfY9NtgggWTOSwNgHN58mQXHN96qwyMgi5yhTM6GtjG3_tXLGeGxX5a5QMkiYsO7
kJrU71wG2-T-uHGEAS2XP4xBCV2b_fOphCbhLdKM4TrXhwULkTQOOpcbL-Cn8YQi7hXUf1YY2vkTUNM9uwAlnOU5OytXw
T-1yEyoD4vViORLQD5OQUI48--VkOKOQom-NrB1O16g8OGIde70YZOOcEx1OwKa64JV9xwdFwwCHaeM9Ctxb2V9oGhROd
QRbSUTPOKnDuIDKi14k7S2_VRYGqDar6oqdKaXTPJ1iaHZbhAw5JRXnZ_Mz6UcO6UyTFBr3Xug;
ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be565f; ApplicationGatewayAffinity=
30860d707537eb845de6189ee6be565f; .AspNetCore.Session=
CfDJ8H5sVbwjamRDoKadqnto%2B25fqfrXyz6biHJBgoxSKdwsn6uIm8wIUEyetUufFh1Ih3B4aeYHE%2BZac6TVdtwdY
T1JKfFQdFr%2F7TvCoPz31gTTVcru5F1BTNOoumsImP1ghRDPjrwyWS%2FPtkpQNwPU21O41SegsupOqTItjwOrLRMI;
Indicium.UseLogSession_48=
CfDJ8H5sVbwjamRDoKadqnto-24Iq6FrylvSFCp8QBfWzruGAczTSN_VOCjrKjrR_CRdx_spTmf7eGb1DGBn4xk2OGVlo
BppLmfWX83zq-L57dCEd9boVoij916B56CoIqN7AtGhnBEFo4NNgYXcinZf-RW_MO_vNXieQKUEaMTNP_4CNYf171Xa8G
qlAgi_1ZYzOJ39iNOyWjCt3inQRGOAVuMcijW7Uq_HD1H52WD6RtxE; Indicium.ApplClaim_48_48=
CfDJ8H5sVbwjamRDoKadqnto-27aXjOO-vypTnka4TIsESBUJ7m7BaWj_XbXsOdxsvxmw-U-TjFtha9SDG-ZKqT3zXDeq
G-_8u3bxNnikiXIRTUGdQfQ9nfib14043N5L_wrFibChr2g1O6sSkBR17Foo9MX5nHY_IbVWCOs8uc811AD1uWski08mk
jLukAwY9vojBCFD26y3ENpCMngN2oOIretwPxtpBvqSFD7SaRQHg8GKLOQFFRJzJeG4xrdWHZfnMqo1dK8JTvDjmVL23r
WNz81R1EII0xWih2Yx5Cx9r1YO9Twq9AKRq7PtWzJGEnxvE66tcX1-N8MafB-VSo4mM8
Content-Length: 344
Cache-Control: max-age=0
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://nsecresilience-prod.thinklab.cloud
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
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=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-Dest: document
Referer:
https://nsecresilience-prod.thinklab.cloud/indicium/account/ui/logout?returnurl=https://nsecr
esilience-prod.thinklab.cloud/universal
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

LogoutID=&IdentityProviderScheme=&ReturnUrl=
https%3A%2F%2Fnsecresilience-prod.thinklab.cloud%2Funiversal&button=logout&
__RequestVerificationToken=
CfDJ8H5sVbwjamRDoKadqnto-249uOYSxhmM2eTL1X3LcxheV1Z-Z2PaLbXN2VPkWh8LguaITENbbcSincL91gmXNt_ix
oi-mGpkfpDQXGN7qVDI62PsFmbuJ4kqwyq-yh1ZSXYmrg-2sqHKHSnCW3GkgDk-T1hPkSti5M6R5nns7rn2K3V_XOGMk-
```

```
HTTP/1.1 302 Found
Date: Tue, 28 Feb 2023 13:37:50 GMT
Content-Length: 0
Connection: close
Server: Microsoft-IIS/10.0
Cache-Control: no-cache,no-store
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Location: https://nsecresilience-prod.thinklab.cloud/universal
Pragma: no-cache
Set-Cookie: LoginChoice=; expires=Thu, 01 Jan 1970 00:00:00 GMT; path=/
Set-Cookie: Indicium.UseLogSession_48=; expires=Thu, 01 Jan 1970 00:00:00 GMT; path=/
Set-Cookie: Indicium.ApplClaim_48_48=; expires=Thu, 01 Jan 1970 00:00:00 GMT; path=/
Set-Cookie: .AspNetCore.Identity.Application=; expires=Thu, 01 Jan 1970 00:00:00 GMT;
path=/indicium; secure; samesite=lax; httponly
Set-Cookie: Identity.External=; expires=Thu, 01 Jan 1970 00:00:00 GMT; path=/indicium;
secure; samesite=lax; httponly
Set-Cookie: Identity.TwoFactorUserId=; expires=Thu, 01 Jan 1970 00:00:00 GMT; path=/indicium;
 secure; samesite=lax; httponly
Set-Cookie: .AspNetCore.Identity.PasswordExpired=; expires=Thu, 01 Jan 1970 00:00:00 GMT;
path=/indicium; secure; samesite=lax; httponly
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=
f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsit
es.net
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self';
```

This means in the backend the session cookie is not immediately invalidated after explicit logout and can still return data that requires authentication. Because the sessions are short-lived and implementation of a fix would be very difficult (and potentially in conflict with leading RFCs) no finding will be included.



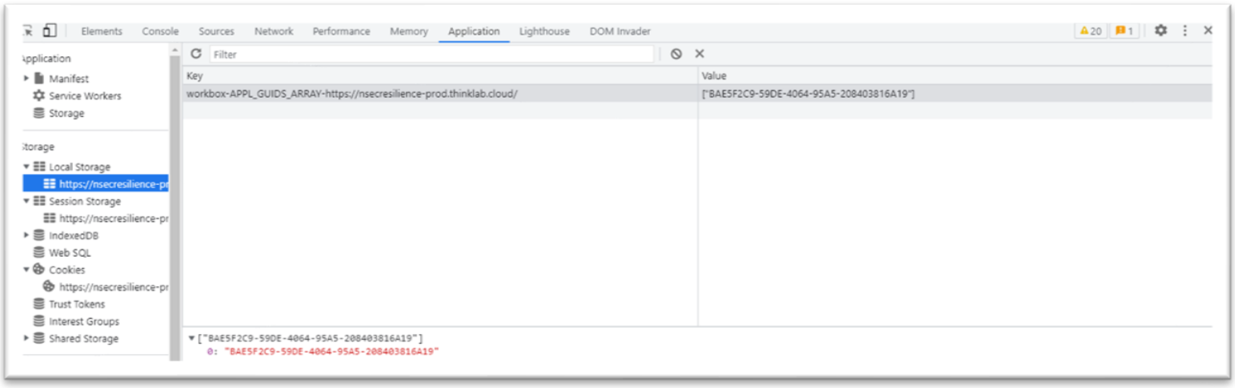
Request

```
GET /indicium/account/ui/dbEventLog HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-251-1yxxkzgcOPsT1PsAQ2_8idqOOLo4pr31h7QyOiEP2VLIPGZ97R7zELoPyTEIKsruRgc2HtFOm4rOdExDXRjf4N5SLvhOJDG-xM_WWuPj_7Zi7sWkeCuzEc9ScnU1DFfesOrrpMTKqO1vRgOCNDZiv1i9KrD6MGeTOvAa
TUoz6bOWzh4QP5UyV8CVj4XfY9NtgggWTOSwNgHN58mQXHN96qwyMg15yhTM6GtjG3_tXLGeGxX5a5QMk1YsO7kJrU71wG2-T-uHGEAS2XP4xBCV2b_fOphCbhLdKM4TrXhwULkTQOOpcbL-Cn8YQ17hXUf1YY2vkTUNM9uwA1nOU5OytXwT-1yEyoD4vViOR
LQD5OQUI48--VkOKOQom-NrB1O16g8OGIde7OYZOOcEx1OwKa64JV9xwdFwwCHaeM9Ctxb2V9oGhROdQRbSUTPOKnDuIDKi14k7S2_VRYGqDar6oqdKaXTPJ1iaHZbhAw5JRXnZ_Mz6UcO6UyTFBr3Xug;
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36
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=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://nsecresilience-prod.thinklab.cloud/universal/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

Response

Jser	Time	Server	Database	Duration	Statement	Message
hinklabadmin	28-02-2023 13:41:03.2740071	nsecresilience-thinklab-sql.germanywestcentral.azurecontainer.io	IAM	2	declare @handle int; exec sp_prepexec @handle output, N'@gui_appl_id int', N'SELECT * FROM i_core_gui_appl WHERE gui_appl_id = @gui_appl_id', @gui_appl_id=48	
hinklabadmin	28-02-2023 13:41:00.0536724	nsecresilience-thinklab-sql.germanywestcentral.azurecontainer.io	INSIGHTS	39	declare @p0 varchar(100) set @p0=null declare @p1 varchar(150) set @p1='AIzaSyAI8-Js9dcX0YfBTuAZ50mkfpqpk4_4W0w' declare @p2 int set @p2=null declare @p3 varchar(max) set @p3=null declare @p4 int set @p4=1 exec prc_save_customer_coordinates_get_customers_without_address @address=@p0 output, @api_key=@p1 output, @customer_id=@p2 output, @response_content=@p3 output,	

Local browser storage was found to be mostly empty, not containing any sensitive or interesting data:



Key	Value
workbox-APPL_GUIDS_ARRAY-https://nsecresilience-prod.thinklab.cloud/	["BAE5F2C9-59DE-4064-95A5-208403816A19"]

```
▼["BAE5F2C9-59DE-4064-95A5-208403816A19"]
   0: "BAE5F2C9-59DE-4064-95A5-208403816A19"
```

No further findings were done with regards to session management.

5. 2 – Broken Access Control

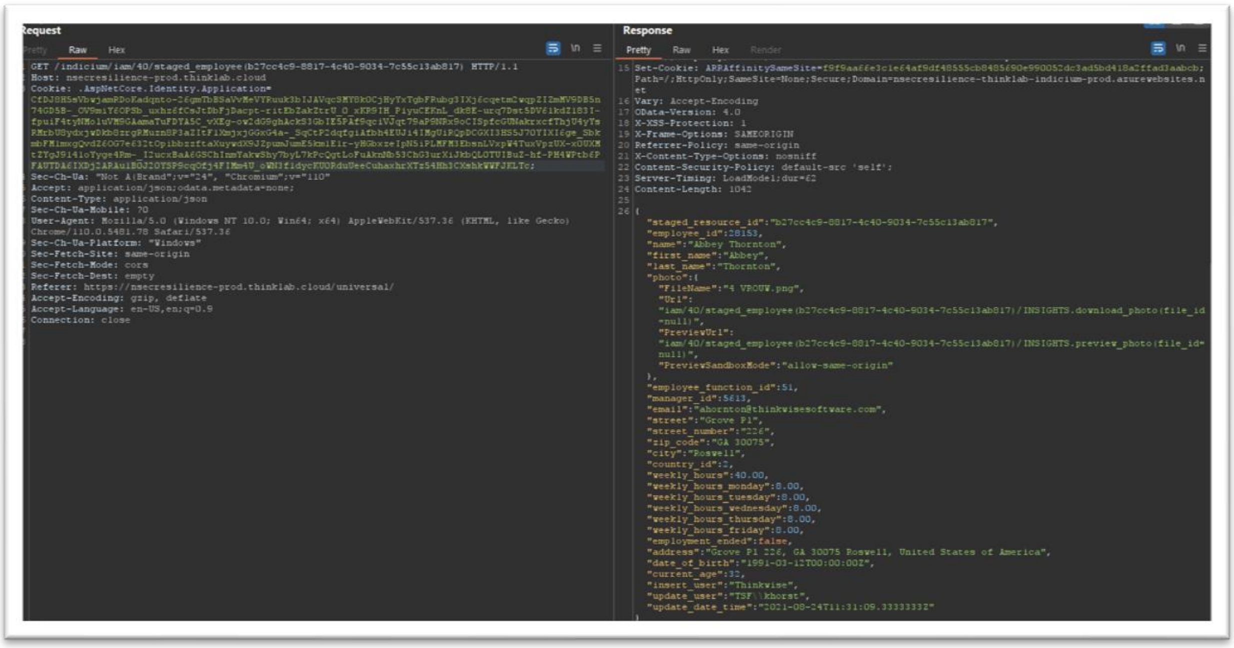
With broken access control, an attacker exploits references to objects or functions to access resources that the attacker is not authorized for. Often this only can be exploited if the authentication management/session management is inadequate. Examples can be references to files with predictable file names, or manipulating function parameters such as an organization ID.

For the Insights example application access control was checked from two perspectives: an attacker without any access to the application, and an attacker with a form of access trying to access functions/data corresponding to a higher access level.

5.2.1 - Access without authentication

Necessary checks where done to understand if an attacker can obtain unauthorized access to functions and/or data that they should not have access to.

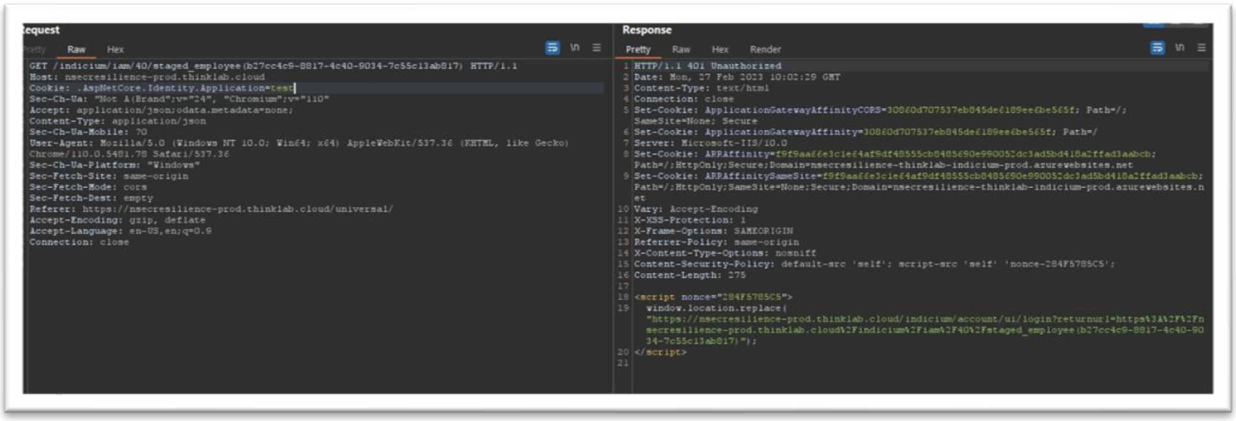
The platform is set up to require a valid *.AspNetCore.Identity.Application* cookie in order to retrieve information.



```
GET /indicium/iam/40/staged_employee(b27cc4c9-8817-4c40-9034-7c55c13ab817) HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-26gmTbBSaVvMeVYRuuk3bIJAVqcSMY8kOCjHyYxTgbFRubg3IXj6cqetm2wqpZI2mMV9DB5n
74GD5B-_OV9miY6OPSb_uxhz6fCsJtDbFjDacpt-ritEbZakZtrU_O_xKP9IH_PiyuCEKnL_dk8E-urq7Dst5DV6ikdZ183I-
fpuiF4tyNMoluVM9GAamaTuFDYA5C_vXEg-ow2dG9ghAckS3GbIE5PAf9qciVJqt79aP9NRx9oCISpfcGUNakrxcfThjU4yYs
RMrbU8ydxjwDkb8zrgRMuzn8P3aZItF1XmjxjGGxG4a-_SqCtP2dqfgiAfbh4EUJi4IMgUiRQpDCGXI3HS5J7OYIXI6ge_Sbk
mbFMlmxgQvdZ6OG7e632tOpibbzzftaXuywdX9JZpumJumE5kmlElr-yHGbxzeIpN5iPLMFM3EbsnLVxpW4TuxVpzUX-xOUXM
tZYgJ9141oYyge4Pm-_I2ucxBaA6GSChImmYakwShy7byL7kPcQgtLoFuAknNb53ChG3urXiJkbQLOTUIBuZ-hf-PH4WPtb6P
FAUTDA6IXDj2APAu1BOJ2OYSP9cqOfj4FIMm4U_oWN3f1dycKUORduUeeCuhaxhrXTz54Hb3CXshkWWFJELTc;
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Accept: application/json;odata.metadata=none;
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://nsecresilience-prod.thinklab.cloud/universal/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

```
Set-Cookie: ARRAffinitySameSite=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.n
et
Vary: Accept-Encoding
OData-Version: 4.0
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self';
Server-Timing: LoadModel;dur=62
Content-Length: 1042

{
  "staged_resource_id":"b27cc4c9-8817-4c40-9034-7c55c13ab817",
  "employee_id":28153,
  "name":"Abbey Thornton",
  "first_name":"Abbey",
  "last_name":"Thornton",
  "photo":{
    "FileName":"4 VROUW.png",
    "Url":
    "iam/40/staged_employee(b27cc4c9-8817-4c40-9034-7c55c13ab817)/INSIGHTS.download_photo(file_id
    =null)",
    "PreviewUrl":
    "iam/40/staged_employee(b27cc4c9-8817-4c40-9034-7c55c13ab817)/INSIGHTS.preview_photo(file_id=
    null)",
    "PreviewSandboxMode":"allow-same-origin"
  },
  "employee_function_id":51,
  "manager_id":5613,
  "email":"ahornton@thinkwisesoftware.com",
  "street":"Grove Pl",
  "street_number":"226",
  "zip_code":"GA 30075",
  "city":"Roswell",
  "country_id":2,
  "weekly_hours":40.00,
  "weekly_hours_monday":8.00,
  "weekly_hours_tuesday":8.00,
  "weekly_hours_wednesday":8.00,
  "weekly_hours_thursday":8.00,
  "weekly_hours_friday":8.00,
  "employment_ended":false,
  "address":"Grove Pl 226, GA 30075 Roswell, United States of America",
  "date_of_birth":"1991-03-12T00:00:00Z",
  "current_age":32,
  "insert_user":"Thinkwise",
  "update_user":"TSF\\khorst",
  "update_date_time":"2021-08-24T11:31:09.3333333Z"
}
```

When this cookie is not present or invalid in a request, a “401 Unauthorized” error is returned as seen on the screenshot below:



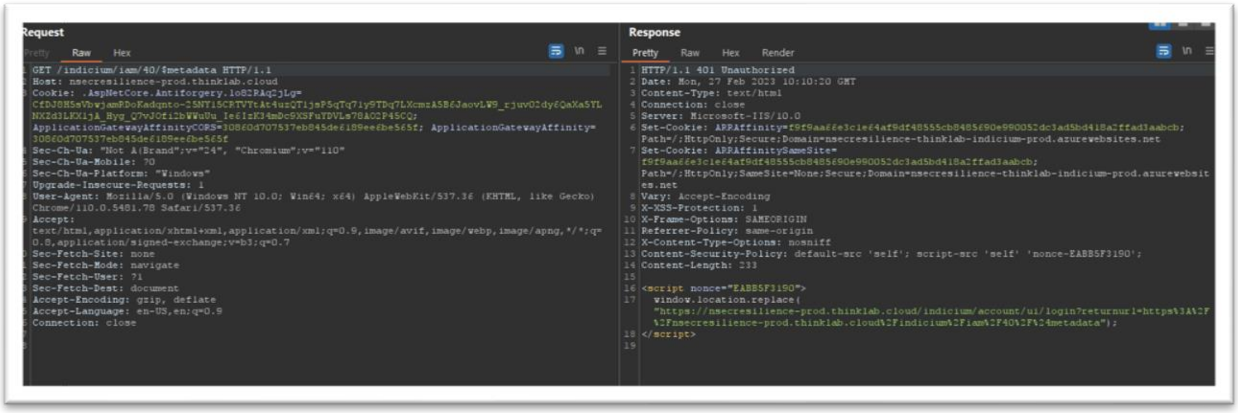
```
GET /indicium/iam/40/staged_employee(b27cc4c9-8817-4c40-9034-7c55c13ab817) HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Identity.Application=test
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Accept: application/json;odata.metadata=none;
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://nsecresilience-prod.thinklab.cloud/universal/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

```
HTTP/1.1 401 Unauthorized
Date: Mon, 27 Feb 2023 10:02:29 GMT
Content-Type: text/html
Connection: close
Set-Cookie: ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be565f; Path=/;
SameSite=None; Secure
Set-Cookie: ApplicationGatewayAffinity=30860d707537eb845de6189ee6be565f; Path=/
Server: Microsoft-IIS/10.0
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.n
et
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self'; script-src 'self' 'nonce-284F5785C5';
Content-Length: 275

<script nonce="284F5785C5">
  window.location.replace(
  "https://nsecresilience-prod.thinklab.cloud/indicium/account/ui/login?returnurl=https%3A%2F%2Fn
  secresilience-prod.thinklab.cloud%2Findicium%2Fiam%2F40%2Fstaged_employee(b27cc4c9-8817-4c40-90
  34-7c55c13ab817)");
</script>
```

This behavior was found to be consistently present; no exceptions were found.

Since the platform makes use of Microsoft's OData standard for the API layer it was also tested if the OData metadata file could be retrieved without authentication. This is relevant because a publicly retrievable metadata file can expose application structure and parameters that could help an attacker find vulnerabilities.

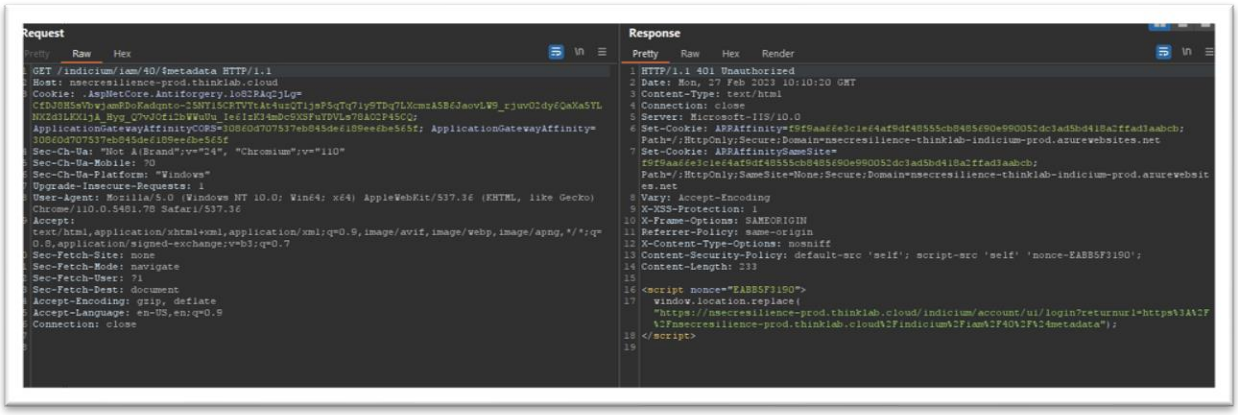
The OData metadata file could however not be retrieved without authentication. Attempts to do so returned a "401 Unauthorized" response, which is good:



```
GET /indicium/iam/40/$metadata HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Antiforgery.1o82RAq2jLg=
CfDJ8H5sVbwjamRDoKadqnto-25NYi5CRTVYtAt4uzQTijsP5qTq7iy9TDq7LXcmzA5B6JaovLW9_rjuv02dy6QaXa5YL
NXZd3LKXijA_Hyg_Q7vJOfi2bWWuUu_Ie6IzK34mDc9XSFuYDVLs78A02P45CQ;
ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be565f; ApplicationGatewayAffinity=
30860d707537eb845de6189ee6be565f
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
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=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

```
HTTP/1.1 401 Unauthorized
Date: Mon, 27 Feb 2023 10:10:20 GMT
Content-Type: text/html
Connection: close
Server: Microsoft-IIS/10.0
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=
f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsit
es.net
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self'; script-src 'self' 'nonce-EABB5F3190';
Content-Length: 233

<script nonce="EABB5F3190">
  window.location.replace(
  "https://nsecresilience-prod.thinklab.cloud/indicium/account/ui/login?returnurl=https%3A%2F
  %2Fnsecresilience-prod.thinklab.cloud%2Findicium%2Fiam%2F40%2F%24metadata");
</script>
```

The same result followed for the OpenAPI documentation which can be used to view available API calls.



Request

Pretty Raw Hex

```
GET /indicium/iam/40/$metadata HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Antiforgery.lo82RAq2jLg=
CfDJ8H5sVbwjamRDoKadqnto-25NYi5CRTVYtAt4uzQTijsP5qTq7iy9TDq7LXcmzA5B6JaovLW9_rjuv02dy6QaXa5YL
NXZd3LKXijA_Hyg_Q7vJOfi2bWWuUu_Ie6IzK34mDc9XSFuYDVLs78A02P45CQ;
ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be565f; ApplicationGatewayAffinity=
30860d707537eb845de6189ee6be565f
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
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=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 401 Unauthorized
Date: Mon, 27 Feb 2023 10:10:20 GMT
Content-Type: text/html
Connection: close
Server: Microsoft-IIS/10.0
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=
f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsit
es.net
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self'; script-src 'self' 'nonce-EABB5F3190';
Content-Length: 233

<script nonce="EABB5F3190">
  window.location.replace(
  "https://nsecresilience-prod.thinklab.cloud/indicium/account/ui/login?returnurl=https%3A%2F
  %2Fnsecresilience-prod.thinklab.cloud%2Findicium%2Fiam%2F40%2F%24metadata");
</script>
```

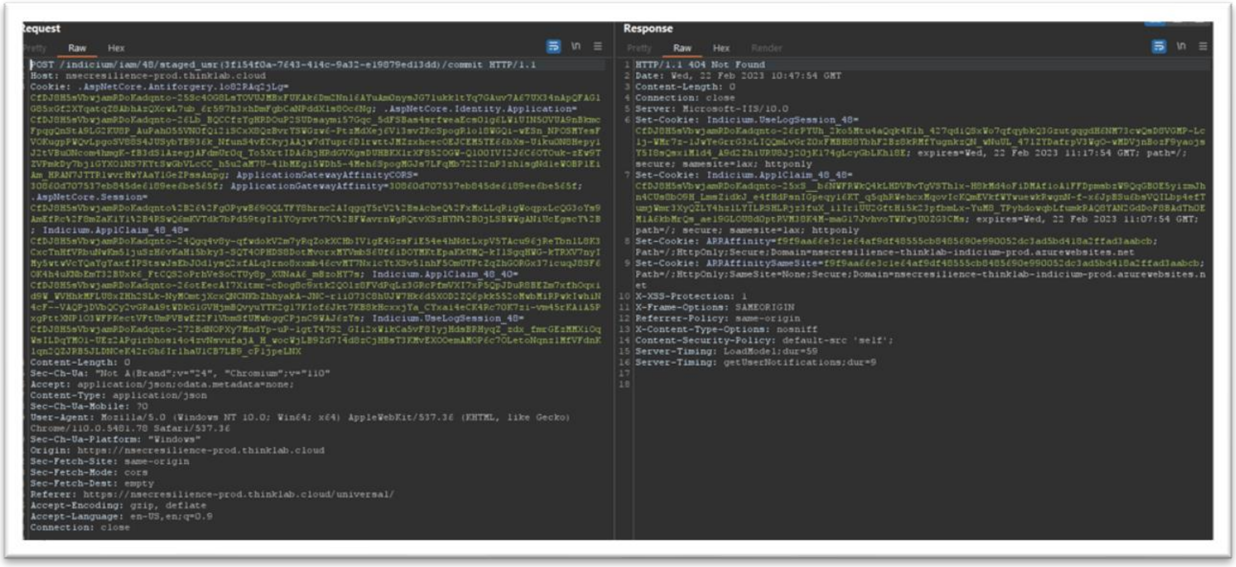
No findings were done for access without authentication.

5.2.2 - Access to functions with higher privilege levels

In this context, the vertical privilege escalation scenarios were investigated. Checks were performed to ensure that access control between different privileges is implemented correctly and consistently.

Various functions/endpoints for the application in-scope were tested across the different roles provided for the applications. As one user has access to the IAM application while the other has not, these could be tested for vertical privilege escalation.

As one of the tests, it was tried to edit the details of another user as low privileged user, by intercepting the request from the IAM to do so and replacing the session cookie:

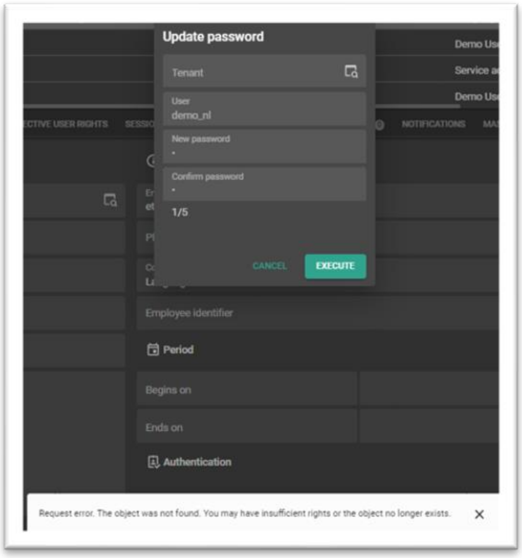


```
POST /indicium/iam/48/staged_usr(3f154f0a-7643-414c-9a32-e19879ed13dd)/commit HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Antiforgery.1o82RAq2jLg=
CfDJ8H5sVbwjamRDoKadqnto-25Sc4OG8LsTOVUJMBxFUKAk6Dm2Nn16AYuAmOnysJG71ukkltYq7GAuv7A67UX34nApQFAG1
G85xGf2XYqatqZ8AbhAzQXcwL7ub_6r597h3xhDmFgbCaNPddXls8Oc6Ng; .AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-26Lb_BQCCfzYgHRDOuP2SUDsaymi57Gqc_5dFSBam4srfweaEcsO1g6LWiUIN5OVUA9nBkmc
FpqgQnStA9LG2KUSP_AuPahO55VNOfQi2iSCxXBQzBvrYSWGzw6-PtzMdXej6Vi3svZRcSpogRlo18WGQi-wESn_NPOSMYesF
VOKugpPWQvLpgoSVB8S4JUSybYB936k_NfunS4vECkyjAAjw7dYupr6DlrwttJMZzxhcecOEJCEMSTE66bXs-UikuONBHepyi
J2tVBuONcom4hmgK-fB3dSiAzegjAFdmUrOq_To5XrtIDA6hjHRdGVXgsDUHBKXlrXFB52OGW-Q1OOIVI2J6C6OTOuk-zEw9T
ZVPmkDy7bj1GYXOiNS7KYtSwGbVLcCC_hSu2aM7U-4lbMEgi5WDh5-4Meh6SpogMGJs7LFqMb722I2nP3zhisgNdieWOBPlEi
Am_HPAN7JTTRlwvrHwYAaY1GeZPssAnpg; ApplicationGatewayAffinityCORS=
3086Od707537eb845de6189ee6be565f; ApplicationGatewayAffinity=3086Od707537eb845de6189ee6be565f;
.AspNetCore.Session=
CfDJ8H5sVbwjamRDoKadqnto%2B26%2FgOPywB69OQLTFY8hrnc2AIqqqY5rV2%2BsAcheQ%2FxMxLLqRigWoqpxLcQG3oYs9
AmEfRc%2F8mZaK1Yi%2B4RSwQ6mKVTdk7bPd59tgIslYOyzvt77C%2BFWavrnWgRQtvXSzHYN%2BOjLSBWWgANiUcEgmcY%2B
; Indicium.ApplClaim_48_48=
CfDJ8H5sVbwjamRDoKadqnto-24Qgq4v8y-qfwdoKVZm7yRqZokXCHbIVigE4GzsFiE54e4hNdtLxpV5TAcu96jReTbnlL8K3
CxcTnHfVRbuNwKm5ljuSzH6vKaHiSbky3-SQT4OPHDS8DotMvorxMYVmbS6Uf6iDOYMKtEpaKkUMQ-kIlSgqHWG-kTRXV7nyI
My5wtWVcYQaYgYaxfIPStswJsEbJOdlysQ2xfALq3rno8xxmb46cvMT7NxicYtXSv5lnhF5OmUYPtZqZhGORGx37icuqJ8SF6
OK4h4uKNbEmT32BUxk6_FtCQSZoPrhVeSoCTUy6p_XUNaA6_mBzoHY7s; Indicium.ApplClaim_48_40=
CfDJ8H5sVbwjamRDoKadqnto-26otEecAI7Xitmr-cDog8c9xtk2QOlz8FVdPqLz3GRcPfmVXI7xP5QpJDuR8HEZm7xfhOqxi
d9W_WVHhkMFLU8xZHh2SLk-NyMOmtjXcxQNCNNbZhhyakA-JNC-rliO73C8hUJW7Hk6d5XOD2ZQ6pkk552oMwbMiRPwkIwhiN
4cF--VAQPjDVbQCy2vGRaA9tWDkGlGVHjmBQvyuYTK2gl7KIof6Jkt7KB8kHcxxjYa_CYxai4eCK4Rc7OK7zi-vm45rKAiASP
xgPttXNP1O3WFPKectVFtUmPVBwEZ2FlVbmSfUMwbggCPjnC9WAJ6zYs; Indicium.UseLogSession_48=
CfDJ8H5sVbwjamRDoKadqnto-272BdNOPXy7MndYp-uP-lgtT47S2_GIi2xWikCa5vF8IyjHdsBRMyqZ_zdx_fmrGEzMMXiOq
WsILDqYMOi-UEz2APgirbhosi4o4zvNsvufajA_H_wocWjLB9Zd7I4d8zCjHBsT3KMvEXOOemAMOP6c7OLetoNqnzlMfVFdnK
1qn2QZJRB5JLDNCeK42rGh6IrlhaUiCB7LB9_cPijpeLNX
Content-Length: 0
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Accept: application/json;odata.metadata=none;
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Origin: https://nsecresilience-prod.thinklab.cloud
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://nsecresilience-prod.thinklab.cloud/universal/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

```
HTTP/1.1 404 Not Found
Date: Wed, 22 Feb 2023 10:47:54 GMT
Content-Length: 0
Connection: close
Server: Microsoft-IIS/10.0
Set-Cookie: Indicium.UseLogSession_48=
CfDJ8H5sVbwjamRDoKadqnto-26rPYUh_2ko5Mtu4aQqk4Kih_427qdiQ5xWo7qfqybkQ3GzutgqgdH6NM73cwQsD8VGMP-Lc
1j-WMr7z-iJwYeGrrG3xLIQQmLvGrZOxFMBHB8YbhFIBz8kRMfYugnkzQN_wNuUL_471ZYDafrpV3WgO-wMDVjnBozF9yaojs
Y5I8sQmxiMld4_A9d2ZhiURU8Jj2OjKi74gLcyGbLKhi8E; expires=Wed, 22 Feb 2023 11:17:54 GMT; path=/;
secure; samesite=lax; httponly
Set-Cookie: Indicium.ApplClaim_48_48=
CfDJ8H5sVbwjamRDoKadqnto-25xS__b6NWFRWkQ4kLHDVBvTgVSThlx-H8kMd4oFiDMAfloAiFFDpmmbzW9QqGBOE5yizmJh
n4CUs8bO9H_LmsZidkJ_e4fHdPsnIGpeqyi6KT_q5qhRWehcxMqovIcKQmEVkfWYwuewkRwgnN-f-x6JpBSu6bsVQILbp4efT
umjWmr3XyQZLY4hzlLYILPSHLRjz3fuX_ilIriUU2GftHi5k23pfbmLx-YuMB_TPyhdowqbLfumkRAQ8YAN2GdDoF8BAdThOE
M1A6kbMrQs_aei9GLOU8dOptRVM38K4M-maGi7JvhvoTWKwjUOZG3CMs; expires=Wed, 22 Feb 2023 11:07:54 GMT;
path=/; secure; samesite=lax; httponly
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.n
et
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self';
Server-Timing: LoadModel;dur=59
Server-Timing: getUserNotifications;dur=9
```

As seen in the screenshot above, the other user does not seem to exist in the context of the low privileged user and a 404 not found error is returned. A 404 error is also returned when a GET request is made to retrieve another user.

Administrators in the IAM application have the ability to change passwords on behalf of a user. It was attempted to intercept the request to change a user's password and change the session cookie to that of a low privileged user (that does not have access to this function) to see if the password of another user could be updated this way. As seen on the screenshot below, updating the password via this function does not require the current password:



Update password

Tenant

User
demo_nl

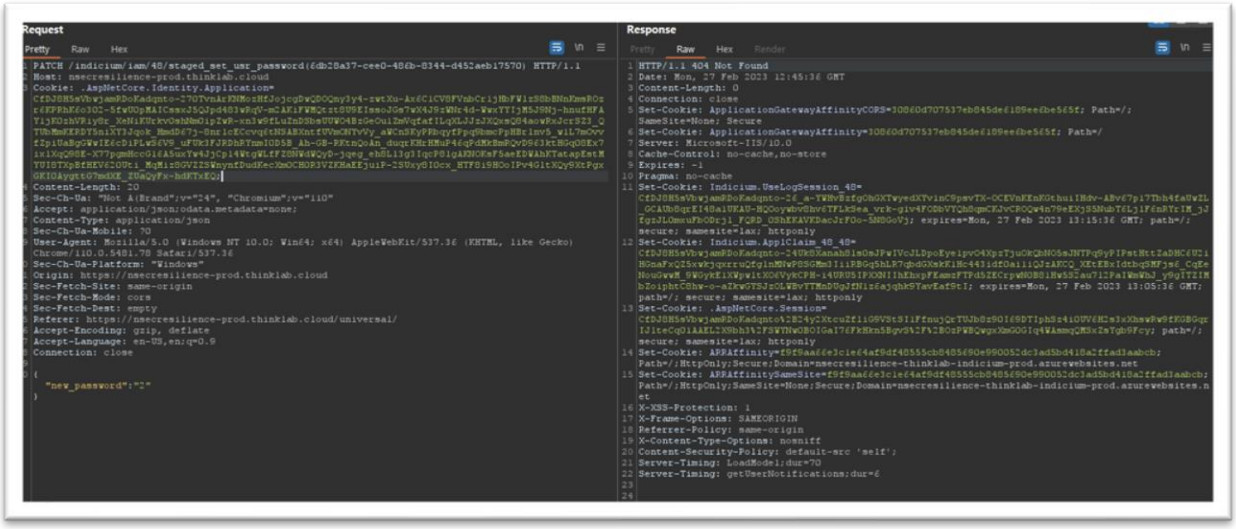
New password
•

Confirm password
•

1/5

Request error. The object was not found. You may have insufficient rights or the object no longer exists.

After replacing the session cookie and sending the request however, a 404 “not found” error is returned, indicating this vertical privilege escalation attack did not succeed which is good.



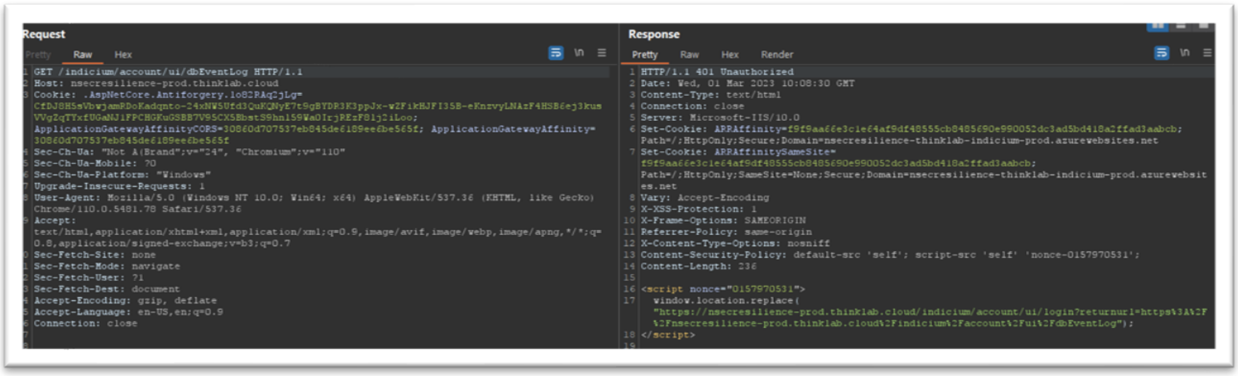
Request

```
PATCH /indicium/iam/48/staged_set_usr_password(6db28a37-cee0-486b-8344-d452aeb17570) HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-270TvnArKNMozHfJojcgDwQDOQny3y4-zwtXu-Ax6C1CV8FVnbCr1jHbFW1zS8bBNnKmmROz
r6KPRhK6o302-5fwUOpMAICsmxJ5QJpd483wRqV-mCAKiFWMQtzt8U9EIsmoJGs7wX4J9zWNr4d-WwxYYIjM5JSNj-hnufHFA
YijKOzhVRiy8r_XeNiKUrkvOshNmOipZwR-xn3w9fLuZnDSbsUUW04BzGeOu1ZmVqfafILqXLJJzJXQxsQ84aowRxJcrSZ3_Q
TUbMmKERDY5niXY3Jqok_HmdD67j-8nr1cECcvq6tNSABXntfUVmCNTvVy_aWCnSKyPRbqyfPpq9bmcPpHBr1mv5_w1L7mOvv
fZpiUaBgGWwIE6cDiPLwS6V9_uFUk3FJRDhRYnmIOD5B_Ah-GB-RKtnQoAn_duqrKHrHMuP46qPdMkBmRQvD963ktHGqO8Ex7
ix1XqQ98E-X77pgmHccG16A5uxYw4JjCp14WtgWLfFZ8NWdWQyD-jqeg_eh8L13g3IqcP81gAKNOKsF5aeEDWAhKTatapEstM
YUI8TXpBfHEV62OUti_MqMiz8GVZZSWnynfDudKecXmOCHOP3VZKHaEEjuiP-2SUxy8IOcx_HTF8i9HOoIPv4G1tXQy9XtPgx
GKIOAygttG7mdXE_ZUaQyFx-hdKTxEQ;
Content-Length: 20
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Accept: application/json;odata.metadata=none;
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Origin: https://nsecresilience-prod.thinklab.cloud
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://nsecresilience-prod.thinklab.cloud/universal/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

{
  "new_password":"2"
}
```

Response

```
HTTP/1.1 404 Not Found
Date: Mon, 27 Feb 2023 12:45:36 GMT
Content-Length: 0
Connection: close
Set-Cookie: ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be565f; Path=/;
SameSite=None; Secure
Set-Cookie: ApplicationGatewayAffinity=30860d707537eb845de6189ee6be565f; Path=/
Server: Microsoft-IIS/10.0
Cache-Control: no-cache,no-store
Expires: -1
Pragma: no-cache
Set-Cookie: Indicium.UseLogSession_48=
CfDJ8H5sVbwjamRDoKadqnto-26_a-YWHvBzfgOhGXTwyedXYvinC9psvTX-OCEVnKEnKGthuiIHdv-ABv67p17Tbh4faUwZL
_GCAUb8qrEI48a1UKAU-HQOoywbv8hv6TFLkSea_vrk-giv4FODbVYQh8qmCKJvCROQw4n79eEXjS5NubT6Lj1F6nRYrIM_jJ
fgzJLOmxuFbODrj1_FQRD_OShEKAVKDacJrFOo-5NBGoVj; expires=Mon, 27 Feb 2023 13:15:36 GMT; path=/;
secure; samesite=lax; httponly
Set-Cookie: Indicium.ApplClaim_48_48=
CfDJ8H5sVbwjamRDoKadqnto-24Uk8Xanah81sOsJPwIVcJLDpoEye1pvO4XpzTjuOkQbNO5sJNTPq9yPIPstHttZaDHC6U2i
HGnaFxQZ5xwkjqxrruQfg1nMNwP8SGMm3IiiRBGq5hLR7qbdGXskKiHc443idfOai1iQJzAKCQ_XEtEBxIdtbqSMFjs6_CqEe
NouGwwM_9WGykE1XWpwitXO6VykCPH-i4URU5IPXXNIIhEhxpFKamzFTPd5ZECrpwNOB81Hw5S2au712PaIWmWhJ_y9gTTZIM
bZoiphtC8hw-o-aZkwGYSJzOLWBvYTMnDUgJfNiz6ajqhk9YavEaf9tI; expires=Mon, 27 Feb 2023 13:05:36 GMT;
path=/; secure; samesite=lax; httponly
Set-Cookie: .AspNetCore.Session=
CfDJ8H5sVbwjamRDoKadqnto%2B24y2XtcuZf1iG9VStSIiFfnujQrTUJb8z9OI69DTIphSz4iOUV6H2s3xXhswRw9fKGBGqr
IJ1teCqO1AAEL2X9bh3%2FSWYNwOBOIGaI76FkHkn5BgvS%2F%2BOzPWBQwgxXmGOGIq4WAmmqQMSxZsYgb9Fcy; path=/;
secure; samesite=lax; httponly
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.n
et
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self';
Server-Timing: LoadModel;dur=70
Server-Timing: getUserNotifications;dur=6
```

Trying to access database event logs as low privileged user, a “401 Unauthorized” error is returned and the user is redirected to login screen:



Request

```
GET /indicium/account/ui/dbEventLog HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Antiforgery.lo82RAq2jLg=
CfDJ8H5sVbwjamRDoKadqnto-24xNW5Ufd3QuKQNyE7t9gBYDR3K3ppJx-wZFikHJFI35B-eKnzvyLNAzF4HSB6ej3kus
VVg2qTYxfUGaNJ1FPCHGKuGSBB7V95CX5BbstS9hnl59Wa0IrjREzF8lj2iLoo;
ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be565f; ApplicationGatewayAffinity=
30860d707537eb845de6189ee6be565f
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
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=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

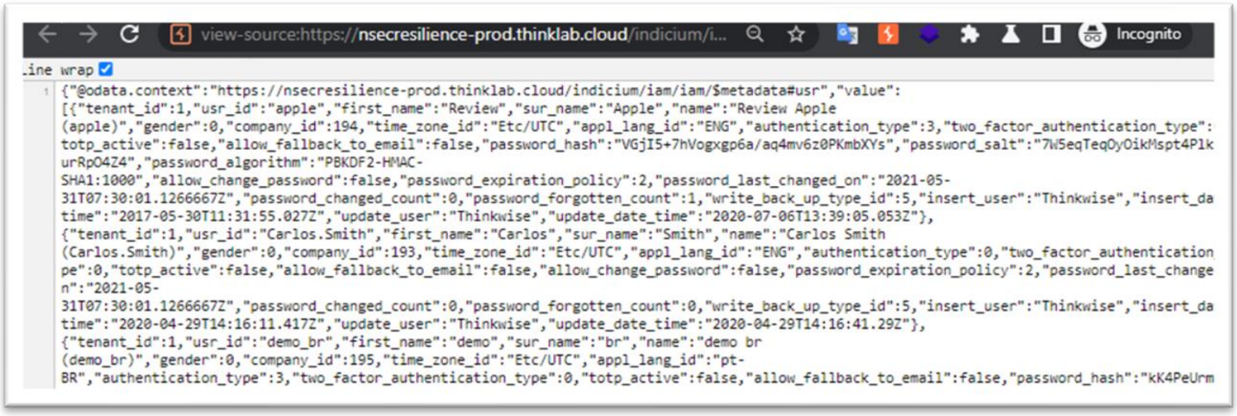
Response

```
HTTP/1.1 401 Unauthorized
Date: Wed, 01 Mar 2023 10:08:30 GMT
Content-Type: text/html
Connection: close
Server: Microsoft-IIS/10.0
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=
f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsit
es.net
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self'; script-src 'self' 'nonce-0157970531';
Content-Length: 236

<script nonce="0157970531">
  window.location.replace(
  "https://nsecresilience-prod.thinklab.cloud/indicium/account/ui/login?returnurl=https%3A%2F
  %2Fnsecresilience-prod.thinklab.cloud%2Findicium%2Faccount%2Fui%2FdbEventLog");
</script>
```

Using the OData metadata file, further checks were done to see if the low privileged user (demo_nl) could access certain service document files that may reveal sensitive information.

For example the IAM admin user can access a service document with sensitive data such as password hashes via:

- nsecresilience-prod.thinklab.cloud/indicium/iam/iam/usr

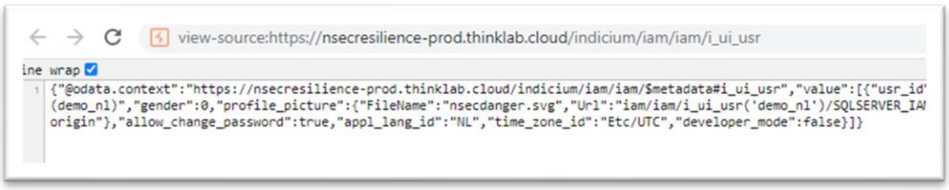


view-source:https://nsecresilience-prod.thinklab.cloud/indicium/i...

Line wrap ✓

```
{"@odata.context":"https://nsecresilience-prod.thinklab.cloud/indicium/iam/iam/$metadata#usr","value":
[{"tenant_id":1,"usr_id":"apple","first_name":"Review","sur_name":"Apple","name":"Review Apple
(apple)","gender":0,"company_id":194,"time_zone_id":"Etc/UTC","appl_lang_id":"ENG","authentication_type":3,"two_factor_authentication_type":
totp_active":false,"allow_fallback_to_email":false,"password_hash":"VGjI5+7hVogxgp6a/aq4mv6z0PKmbXYs","password_salt":"7W5eqTeqOyOikMspt4Plk
urRpO4Z4","password_algorithm":"PBKDF2-HMAC-
SHA1:1000","allow_change_password":false,"password_expiration_policy":2,"password_last_changed_on":"2021-05-
31T07:30:01.1266667Z","password_changed_count":0,"password_forgotten_count":1,"write_back_up_type_id":5,"insert_user":"Thinkwise","insert_da
time":"2017-05-30T11:31:55.027Z","update_user":"Thinkwise","update_date_time":"2020-07-06T13:39:05.053Z"},
{"tenant_id":1,"usr_id":"Carlos.Smith","first_name":"Carlos","sur_name":"Smith","name":"Carlos Smith
(Carlos.Smith)","gender":0,"company_id":193,"time_zone_id":"Etc/UTC","appl_lang_id":"ENG","authentication_type":0,"two_factor_authentication
pe":0,"totp_active":false,"allow_fallback_to_email":false,"allow_change_password":false,"password_expiration_policy":2,"password_last_change
n":"2021-05-
31T07:30:01.1266667Z","password_changed_count":0,"password_forgotten_count":0,"write_back_up_type_id":5,"insert_user":"Thinkwise","insert_da
time":"2020-04-29T14:16:11.417Z","update_user":"Thinkwise","update_date_time":"2020-04-29T14:16:41.29Z"},
{"tenant_id":1,"usr_id":"demo_br","first_name":"demo","sur_name":"br","name":"demo br
(demo_br)","gender":0,"company_id":195,"time_zone_id":"Etc/UTC","appl_lang_id":"pt-
BR","authentication_type":3,"two_factor_authentication_type":0,"totp_active":false,"allow_fallback_to_email":false,"password_hash":"kK4PeUrm
```

When trying to access this as the low privileged user, a 404 “not found” error is returned. The low privileged user could only retrieve his/her own data, not including any password hashes:



view-source:https://nsecresilience-prod.thinklab.cloud/indicium/iam/iam/i_ui_usr

ine wrap ✓

```
{"@odata.context":"https://nsecresilience-prod.thinklab.cloud/indicium/iam/iam/$metadata#i_ui_usr","value":[{"usr_id'
(demo_nl)","gender":0,"profile_picture":{"FileName":"nsecdanger.svg","Url":"iam/iam/i_ui_usr('demo_nl')/SQLSERVER_IAI
origin"},"allow_change_password":true,"appl_lang_id":"NL","time_zone_id":"Etc/UTC","developer_mode":false}]}
```

No findings were done with regards to vertical privilege escalation.

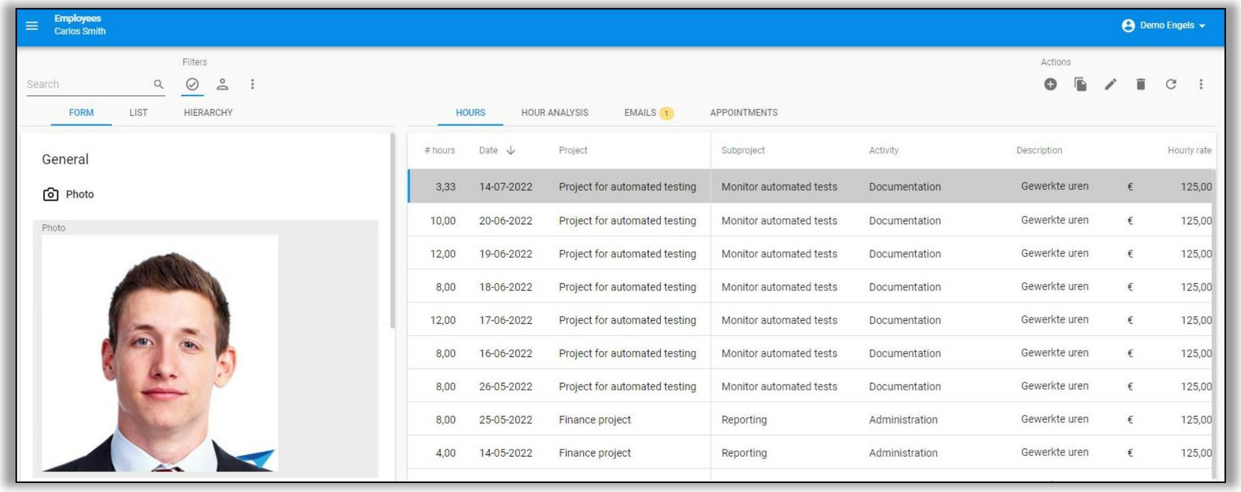
5.3 – Unrestricted File Upload

With unrestricted file upload, an attacker uses the functionality to upload files. Instead of valid files other vulnerable files are uploaded. For example, a remote shell can be uploaded so that control can be taken on the server.

For file upload checks focus was placed on the INSIGHTS application because the upload functions that were present there are also available to normal end users (unlike the upload functions in the IAM application, which only administrators can access).

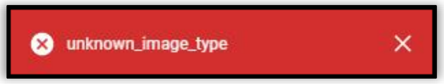
First, testers identified the unique upload features present in the INSIGHTS application.

In the Insight application photos can be uploaded of an individual. Several tests have been performed on this upload function. The first test was to upload files with potentially dangerous file extensions, such as .exe, .html, etc.



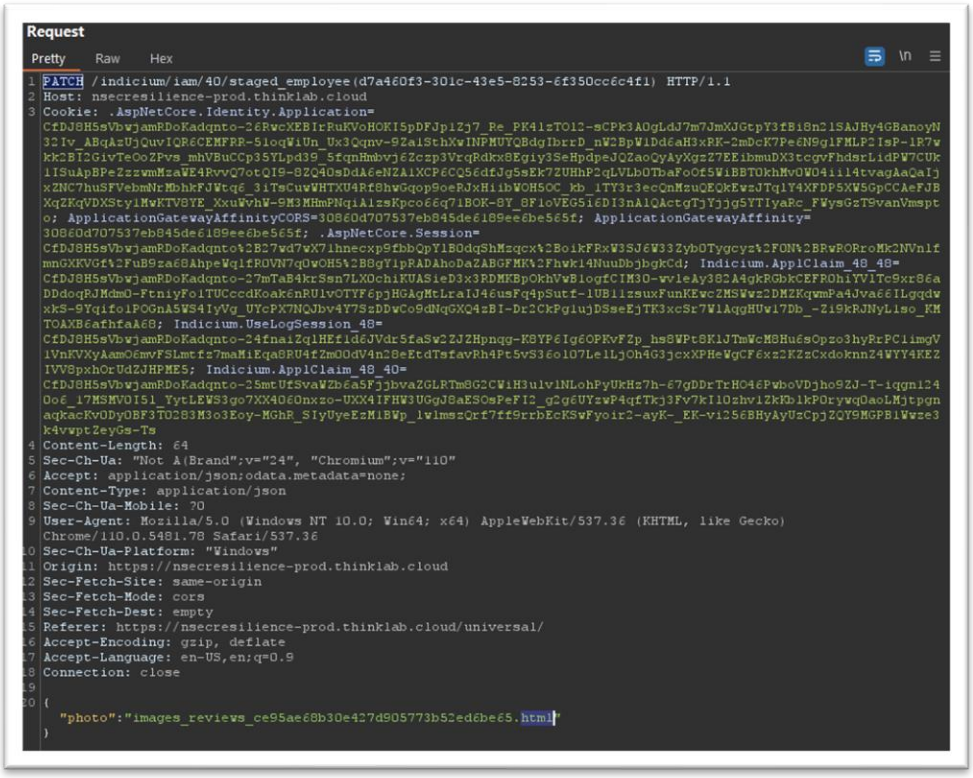
# hours	Date	Project	Subproject	Activity	Description	Hourly rate
3,33	14-07-2022	Project for automated testing	Monitor automated tests	Documentation	Gewerkte uren	€ 125,00
10,00	20-06-2022	Project for automated testing	Monitor automated tests	Documentation	Gewerkte uren	€ 125,00
12,00	19-06-2022	Project for automated testing	Monitor automated tests	Documentation	Gewerkte uren	€ 125,00
8,00	18-06-2022	Project for automated testing	Monitor automated tests	Documentation	Gewerkte uren	€ 125,00
12,00	17-06-2022	Project for automated testing	Monitor automated tests	Documentation	Gewerkte uren	€ 125,00
8,00	16-06-2022	Project for automated testing	Monitor automated tests	Documentation	Gewerkte uren	€ 125,00
8,00	26-05-2022	Project for automated testing	Monitor automated tests	Documentation	Gewerkte uren	€ 125,00
8,00	25-05-2022	Finance project	Reporting	Administration	Gewerkte uren	€ 125,00
4,00	14-05-2022	Finance project	Reporting	Administration	Gewerkte uren	€ 125,00

When a file with a disallowed file extension was selected in the UI, the following error was displayed:


unknown_image_type

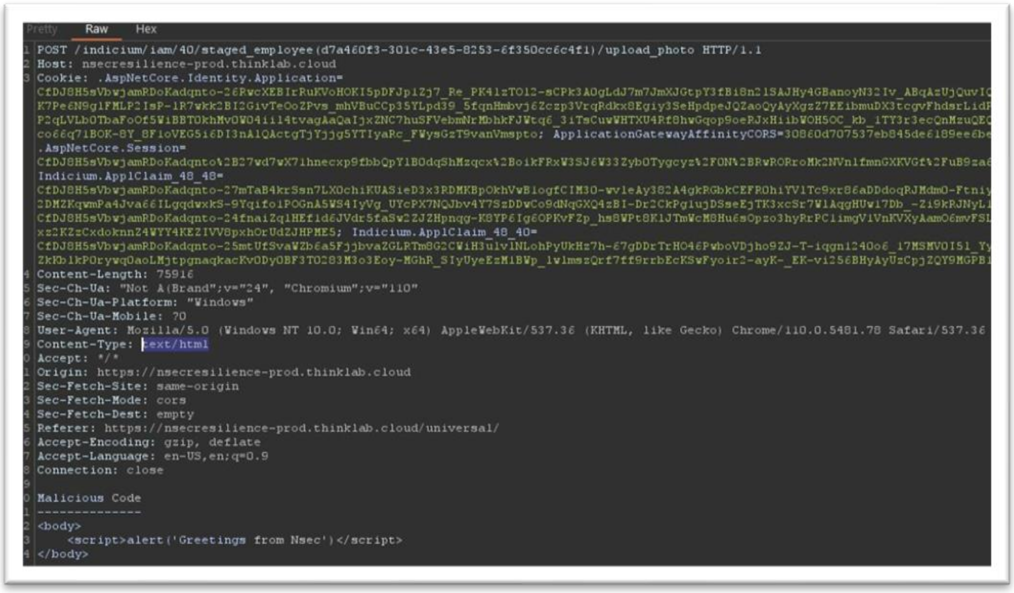
It was not possible from the frontend/UI to select a file with an extension other than image file extensions, like jpeg, jpg, png, etc.

However, it was possible to intercept the PATCH request (that makes up the first step in the upload process) and change the file extension from (for example) .png to .html:



```
PATCH /indicium/iam/40/staged_employee(d7a460f3-301c-43e5-8253-6f350cc6c4f1) HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-26RwcXEBIrRuKVoHOKI5pDFJp1Zj7_Re_PK41zTO12-sCPk3AOgLdJ7m7JmXJGtpY3fBi8n21SAJHy4GBanoyN
32Iv_ABqAzUjQuvIQR6CEMFRR-51oqWiUn_Ux3Qqnv-9Za1SthXwINPMUYQBdgIbrrD_nW2BpW1Dd6aH3xRK-2mDcK7Pe6N9g1FMLP2IsP-1R7w
kk2BI2GivTeOoZPvs_mhVBuCCp35YLpd39_5fqnHmbvj6Zczp3VrqRdkx8Egiy3SeHpdpeJQZaoQyAyXgzZ7EEibmuDX3tcgvFhdsrLidPW7CUk
1ISuApBPeZzzwmMzaWE4RvvQ7otQI9-8ZQ40sDdA6eNZA1XCP6CQ56dfJg5sEk7ZUHhP2qLVLbOTbaFoOf5WiBBTOkhMvOW04iil4tvagAaQaIj
xZNC7huSFVebmNrMbhkFJWtq6_3iTsCuwWHTXU4Rf8hwGqop9oeRJxHiibWOH5OC_kb_1TY3r3ecQnMzuQEQkEwzJTq1Y4XFDP5XW5GpCCAeFJB
XqZKqVDXSty1MwKTV8YE_XxuWvhW-9M3MHmPNqiA1zsKpco66q71BOK-8Y_8F1oVEG5i6DI3nA1QActgTjYjjg5YTIyaRc_FWysGzT9vanVmspt
o; ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be565f; ApplicationGatewayAffinity=
30860d707537eb845de6189ee6be565f; .AspNetCore.Session=
CfDJ8H5sVbwjamRDoKadqnto%2B27wd7wX71hnecxp9fbbQpY1BOdqShMzqcx%2BoikFRxW3SJ6W33ZybOTygcyz%2FON%2BRwRORroMk2NVnlf
mnGXKVGf%2FuB9za68AhpeWq1fROVN7qOwOH5%2B8gY1pRADAhoDaZABGFMK%2Fhwk14NuuDbjbgkCd; Indicium.ApplClaim_48_48=
CfDJ8H5sVbwjamRDoKadqnto-27mTaB4krSsn7LXOchiKUASieD3x3RDMKBpOkhVwB1ogfCIM30-wvleAy382A4gkRGbkCEFROhiYV1Tc9xr86a
DDdoqRJMdm0-FtniyFo1TUCccdKoak6nRU1vOTYF6pjHGAgMtLraIJ46usFq4pSutf-1UB1lzsuxFunKEwcZMSWwz2DMZKqwmPa4Jva66ILgqdw
xkS-9Yqifo1POGnA5WS4IyVg_UYcPX7NQJbv4Y7SzDDwCo9dNqGXQ4zBI-Dr2CkPg1ujDSseEjTK3xcSr7W1AqgHUw17Db_-Zi9kRJNyL1so_KM
TOAXB6afhfaA68; Indicium.UseLogSession_48=
CfDJ8H5sVbwjamRDoKadqnto-24fnaiZq1HEf1d6JVdr5faSw2ZJZHpnqg-K8YP6Ig6OPKvFZp_hs8WPt8K1JTmWcM8Hu6sOpzo3hyRrPC1imgV
1VnKVXyAamO6mvFSLmtfz7maMiEqa8RU4fZmO0dV4n28eEtdTsfavRh4Pt5vS36o107Le1LjOh4G3jcxXPHeWgCF6xz2KZzCxdoknnZ4WYY4KEZ
IVV8pxhOrUdZJHPME5; Indicium.ApplClaim_48_40=
CfDJ8H5sVbwjamRDoKadqnto-25mtUfSvaWZb6a5FjjbvaZGLRTm8G2CWiH3ulv1NLohPyUkHz7h-67gDDrTrHO46PwboVDjho9ZJ-T-iqgn124
Oo6_17MSMVOI51_YytLEWS3go7XX4060nxzo-UXX4IFHW3UGgJ8aESOsPeFI2_g2g6UYzwP4qfTkj3Fv7kI10zhv1ZkKb1kPOrywqOaoLMjtpgn
aqkacKvODyOBF3TO283M3o3Eoy-MGhR_SIyUyeEzM1BWp_1w1mszQrf7ff9rrbEcKSwFyoir2-ayK-_EK-vi256BHyAyUzCpjZQY9MGPB1Wwze3
k4vwptZeyGs-Ts
Content-Length: 64
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Accept: application/json;odata.metadata=none;
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Origin: https://nsecresilience-prod.thinklab.cloud
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://nsecresilience-prod.thinklab.cloud/universal/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

{
  "photo":"images_reviews_ce95ae68b30e427d905773b52ed6be65.html"
}
```

After this, the second upload request (a POST message) was also intercepted. The Content-Type was changed to text/html and the request was sent:



```
POST /indicium/iam/40/staged_employee(d7a460f3-301c-43e5-8253-6f350cc6c4f1)/upload_photo HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-26RwcXEBIrRuKVoHOKI5pDFJp1Zj7_Re_PK41zTO12-sCPk3AOgLdJ7m7JmXJGtpY3fBi8n21SAJHy4GBanoyN32Iv_ABqAzUjQuvIQ
K7Pe6N9g1FMLP2IsP-1R7wkk2BI2GivTeOoZPvs_mhVBuCCp35YLpd39_5fqnHmbvj6Zczp3VrqRdkx8Egiy3SeHpdpeJQZaoQyAyXgzZ7EEibmuDX3tcgvFhdsrLidP
P2qLVLbOTbaFoOf5WiBBTOkhMvOW04iil4tvagAaQaIjxZNC7huSFVebmNrMbhkFJWtq6_3iTsCuwWHTXU4Rf8hwGqop9oeRJxHiibWOH5OC_kb_1TY3r3ecQnMzuQEQ
co66q71BOK-8Y_8F1oVEG5i6DI3nA1QActgTjYjjg5YTIyaRc_FWysGzT9vanVmspto; ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be
.AspNetCore.Session=
CfDJ8H5sVbwjamRDoKadqnto%2B27wd7wX71hnecxp9fbbQpY1BOdqShMzqcx%2BoikFRxW3SJ6W33ZybOTygcyz%2FON%2BRwRORroMk2NVnlfmnGXKVGf%2FuB9za6
Indicium.ApplClaim_48_48=
CfDJ8H5sVbwjamRDoKadqnto-27mTaB4krSsn7LXOchiKUASieD3x3RDMKBpOkhVwB1ogfCIM30-wvleAy382A4gkRGbkCEFROhiYV1Tc9xr86aDDdoqRJMdm0-Ftniy
2DMZKqwmPa4Jva66ILgqdwxkS-9Yqifo1POGnA5WS4IyVg_UYcPX7NQJbv4Y7SzDDwCo9dNqGXQ4zBI-Dr2CkPg1ujDSseEjTK3xcSr7W1AqgHUw17Db_-Zi9kRJNyL1
CfDJ8H5sVbwjamRDoKadqnto-24fnaiZq1HEf1d6JVdr5faSw2ZJZHpnqg-K8YP6Ig6OPKvFZp_hs8WPt8K1JTmWcM8Hu6sOpzo3hyRrPC1imgV1VnKVXyAamO6mvFSL
xz2KZzCxdoknnZ4WYY4KEZIVV8pxhOrUdZJHPME5; Indicium.ApplClaim_48_40=
CfDJ8H5sVbwjamRDoKadqnto-25mtUfSvaWZb6a5FjjbvaZGLRTm8G2CWiH3ulv1NLohPyUkHz7h-67gDDrTrHO46PwboVDjho9ZJ-T-iqgn124Oo6_17MSMVOI51_Yy
ZkKb1kPOrywqOaoLMjtpgnaqkacKvODyOBF3TO283M3o3Eoy-MGhR_SIyUyeEzM1BWp_1w1mszQrf7ff9rrbEcKSwFyoir2-ayK-_EK-vi256BHyAyUzCpjZQY9MGPB1
Content-Length: 75916
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Sec-Ch-Ua-Platform: "Windows"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36
Content-Type: text/html
Accept: */*
Origin: https://nsecresilience-prod.thinklab.cloud
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://nsecresilience-prod.thinklab.cloud/universal/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

Malicious Code
--------------
<body>
    <script>alert('Greetings from Nsec')</script>
</body>
```

After these changes were applied, the request was sent to the backend server and the .html file was successfully uploaded. This file was then temporarily available through a URL that could be accessed, for example:

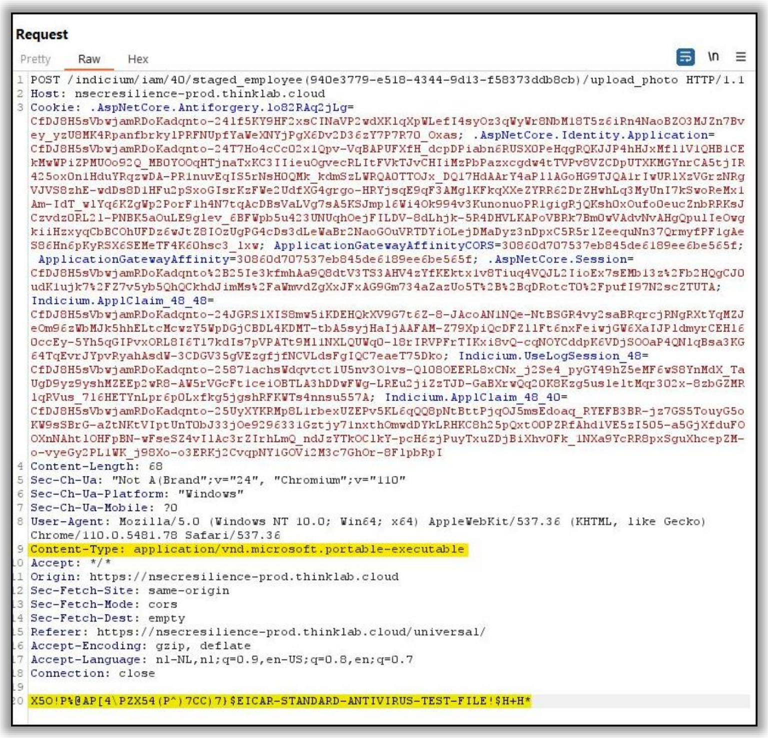
- https://nsecresilience-prod.thinklab.cloud/indicium/iam/40/staged_employee(d7a460f3-301c-43e5-8253-6f350cc6c4f1)/INSIGHTS.download_photo(file_id=null)?t=1677244462911

Once upload has been completed and the uploaded file is stored on the backend server it cannot be opened in the context of the web application itself for example as part of a preview function. The file is immediately downloaded locally and can then be opened in a browser of choice. This mitigates most of the direct risk; if files with dangerous extensions could be made to be rendered directly by the webserver, this could lead to code execution.

Checks in file upload functions require validations for file extensions on both the frontend and backend. On the Insights application this was only the case on the frontend. However for Thinkwise in general allowed file extensions can be configured in the Software Factory ni the ‘extention whitelist’ setting that can be set for each ‘file storage location’. Because of this possibility no finding will be included.

In addition to tests on validations of file extensions, tests were also performed with the goal of establishing whether there is an active virus scanner active on the file upload functionalities. For this purpose a so-called EICAR file was used. The EICAR Standard Anti-Malware Test file is a special 'dummy' file which is used to test the correct operation of malware detection scanners. When an EICAR test file is placed on a file system, any virus scanner that is active on that file system will detect it exactly as if it were a malicious program. Alternatively, some virus scanners can check file contents as part of the upload function itself, and block files before they are placed on the operating system.

In the case of the Insights application the EICAR file was uploaded without any restrictions:



```
POST /indicium/iam/40/staged_employee(940e3779-e518-4344-9d13-f58373ddb8cb)/upload_photo HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Antiforgery.lo82RAq2jLg=
CfDJ8H5sVbwjamRDoKadqnto-241f5KY9HF2xsCINaVP2wdXK1qXpWLefI4syOz3qWyWr8NbM18T5z6iRn4NaoBZO3MJZn7Bv
ey_yzU8MK4RpanfbrkylPRFNUpfYaWeXNYjPgX6Dv2D36zY7P7R70_Oxas; .AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-24T7Ho4cCc02x1Qpv-VqBAPUFXfH_dcpDPiabn6RUSXOPeHqgRQKJJP4hHJxMf11V1QHB1CE
kMwWPiZPMUOo92Q_MBOYOOqHTjnaTxKC3IIieuOgvecRLItFVkTJvCHIiMzPbPazxcgdw4tTVPv8VZCDpUTXKMGYnrCA5tjIR
425oxOn1HduYRqzwDA-PR1nuvEqIS5rNsHOQMk_kdmSzLWRQAOTTOJx_DQ17HdAArY4aP11AGoHG9TJQA1rIwUR1XzVGrzNRg
VJVS8zhE-wdDs8D1HFu2pSxoGIsrKzFWe2UdfXG4grgo-HRYjsqE9qF3AMg1KFkqXXeZYRR62DrZHwhLq3MyUnI7kSwoReMx1
Am-IdT_w1Yq6KZgWp2PorF1h4N7tqAcDBsVaLVg7sA5KSJmp16Wi4Ok994v3KunonuoPR1gigRjQKshOxOufoOeucZnbRRKsJ
CzvdzORL21-PNBK5aOuLE9glev_6BFWpb5u423UNUqhOejFILDV-8dLhjk-5R4DHVLKAPoVBRk7BmOwVAdvNvAHgQpulIeOwg
kiiHzxyqCbBCOhUFDz6wJtZ8IOzUgPG4cDs3dLeWaBr2NaoGOuVRTDYiOLejDMaDyz3nDpxC5R5r1ZeequNn37QrmyfPF1gAe
S86Hn6pKyRSX6SEMeTF4K6Ohsc3_1xw; ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be565f;
 ApplicationGatewayAffinity=30860d707537eb845de6189ee6be565f; .AspNetCore.Session=
CfDJ8H5sVbwjamRDoKadqnto%2B25Ie3kfmhAa9Q8dtV3TS3AHV4zYfKEktx1v8Tiuq4VQJL2IioEx7sEMb13z%2Fb2HQgCJO
udK1ujk7%2FZ7v5yb5QhQCkhdJimMs%2FaWmvdZgXxJFxAG9Gm734aZazUo5T%2B%2BqDRotcTO%2FpufI97N2scZTUTA;
Indicium.ApplClaim_48_48=
CfDJ8H5sVbwjamRDoKadqnto-24JGRS1XIS8mw5iKDEHQkXV9G7t6Z-8-JAcoAN1NQe-NtBSGR4vy2saBRqrcjRNgRXtYqMZJ
eOm96zWbMJk5hhELtcMcwzY5WpDGjCBDL4KDMT-tbA5syjHaIjAAFAM-Z79XpiQcDFZ11Ft6nxFeiwjGW6XaIJP1dmyrCEH16
OccEy-5Yh5qGIPvxORL8I6T17kdIs7pVPATt9M11NXLQUWqO-18rIRVPFrTIKxi8vQ-cqNOYCddpK6VDjSOOaP4QN1qBsa3KG
64TqEvrJYpvRyahAsdW-3CDGV35gVEzgfjfNCVLdsFgIQC7eaeT75Dko; Indicium.UseLogSession_48=
CfDJ8H5sVbwjamRDoKadqnto-25871achsWdqvtct1U5nv3O1vs-Q1O8OEERL8xCNx_j2Se4_pyGY49hZ5eMF6wS8YnMdX_Ta
UgD9yz9yshMZEEp2wR8-AW5rVGcFt1ceiOBTLA3hDDwFWg-LREu2jiZzTJD-GaBXrwQq2OK8Kzg5usle1tMqr3O2x-8zbGZMR
1qRVus_716HETYnLpr6pOLxfkg5jgshRFKWTs4nnsu557A; Indicium.ApplClaim_48_40=
CfDJ8H5sVbwjamRDoKadqnto-25UyXYKRMp8L1rbexUZEPv5KL6qQQ8pNtBttPjqOJ5msEdoaq_RYEFB3BR-jz7GS5TouyG5o
KW9sSBrG-aZtNKtVIptUnTObJ33jOe929633lGztjy71nxthOmwdDYkLRHKC8h25pQxtOOPZRfAhd1VE5zI5O5-a5GjXfduFO
OXnNAht1OHFpBN-wFseSZ4vI1Ac3rZIrhLmQ_ndJzYTkOC1kY-pcH6zjPuyTxuZDjBiXhvOFk_1NXa9YcRR8pxSguXhcepZM-
o-vyeGy2PL1WK_j98Xo-o3ERKj2CvqpNY1GOVi2M3c7GhOr-8F1pbRpI
Content-Length: 68
Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium";v="110"
Sec-Ch-Ua-Platform: "Windows"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.78 Safari/537.36
Content-Type: application/vnd.microsoft.portable-executable
Accept: */*
Origin: https://nsecresilience-prod.thinklab.cloud
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://nsecresilience-prod.thinklab.cloud/universal/
Accept-Encoding: gzip, deflate
Accept-Language: nl-NL,nl;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

Since the EICAR file was not blocked by the webserver it did not seem that any virus scanner is active on the location where these files are saved. Since the file after saving the data, with the HR employees for example, is immediately downloaded locally into the client's browser, this can bring a risk with it. In general the advice is to make sure that there is a virus scanner active on the file system to which files are uploaded. Because the implementation of security measures in this form are the responsibility of partner organizations or end users, no finding will be included for this.

5.4 – Directory traversal and Remote file inclusion

With directory traversal it is possible to change the URL of the web application in such a way that files in other directories (outside the location of the files of the web application) can be accessed. With remote file inclusion same thing happens but this is done through including remote files through file parameters in functions.

Moderate manual testing was done on regular directory traversal based on windows directories and files, respectively in both regular format as encoded format. As automated tool, next to NetSparker and OWASP ZAP also dotdotpwn was used. No vulnerabilities were identified. For file inclusion no attack vectors were identified.

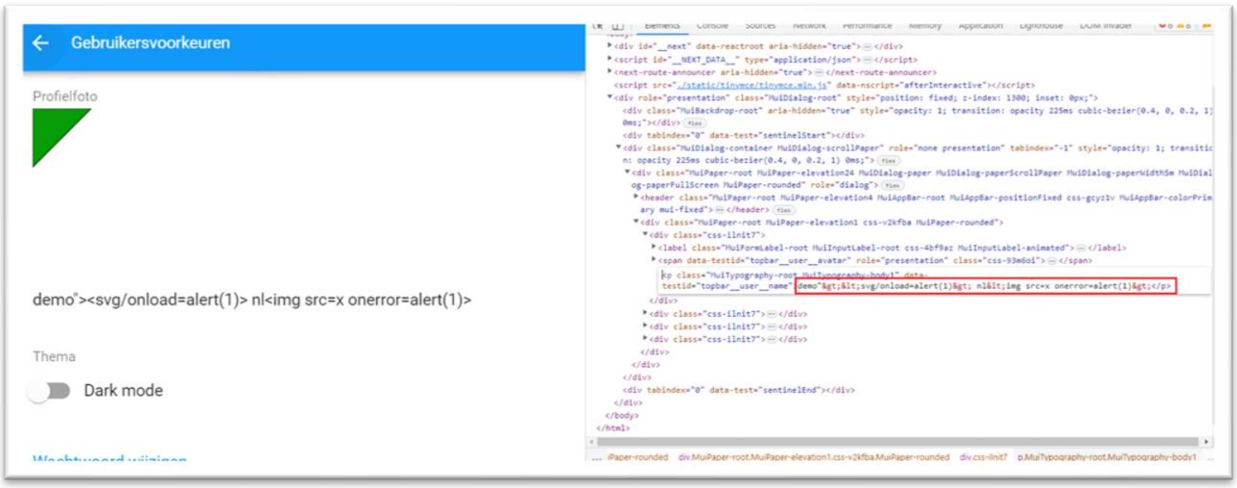
5.5 – Cross-site scripting (XSS)

Cross-site scripting is a technique in which an attacker makes use of lack of sanitation of user input. The attacker tries to leverage such as vulnerability by identifying a place where any malicious input will be presented back to a user. The attacker then inputs malicious code, for example to steal cookies and send them to the attacker, and waits for another user to trigger the scripts. That way, the attacker can collect information or even control complete user browser sessions.

During the penetration test various forms and functions with user input were tested for cross-site scripting. Automated scans were used to test for reflected cross-site scripting extensively. Stored XSS was investigated mainly using manual testing.

It was found that generally html/javascript injection attempts are blocked by consistently by the application by applying entity encoding.

For example, using the admin IAM account, XSS payloads were injected inside the username of demo_nl user, as the name field is referenced often within the application.

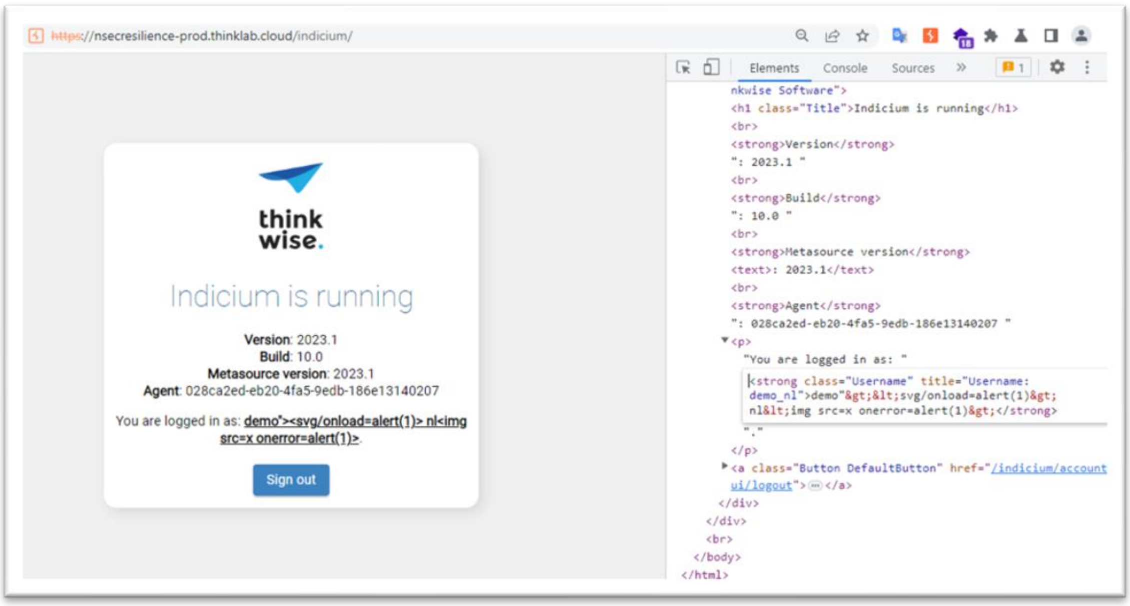


demo"><svg/onload=alert(1)> nl<img src=x onerror=alert(1)>

```
▸<div id="__next" data-reactroot aria-hidden="true">⋯</div>
▸<script id="__NEXT_DATA__" type="application/json">⋯</script>
▸<next-route-announcer aria-hidden="true">⋯</next-route-announcer>
 <script src="./static/tinymce/tinymce.min.js" data-nscript="afterInteractive"></script>
▾<div role="presentation" class="MuiDialog-root" style="position: fixed; z-index: 1300; inset: 0px;">
   <div class="MuiBackdrop-root" aria-hidden="true" style="opacity: 1; transition: opacity 225ms cubic-bezier(0.4, 0, 0.2, 1)
   0ms;"></div> flex
   <div tabindex="0" data-test="sentinelStart"></div>
  ▾<div class="MuiDialog-container MuiDialog-scrollPaper" role="none presentation" tabindex="-1" style="opacity: 1; transitic
   n: opacity 225ms cubic-bezier(0.4, 0, 0.2, 1) 0ms;"> flex
   ▾<div class="MuiPaper-root MuiPaper-elevation24 MuiDialog-paper MuiDialog-paperScrollPaper MuiDialog-paperWidthSm MuiDial
    og-paperFullScreen MuiPaper-rounded" role="dialog"> flex
    ▸<header class="MuiPaper-root MuiPaper-elevation4 MuiAppBar-root MuiAppBar-positionFixed css-gcyz1v MuiAppBar-colorPrim
     ary mui-fixed">⋯</header> flex
    ▾<div class="MuiPaper-root MuiPaper-elevation1 css-v2kfba MuiPaper-rounded">
     ▾<div class="css-ilnit7">
       ▸<label class="MuiFormLabel-root MuiInputLabel-root css-4bf9az MuiInputLabel-animated">⋯</label>
       ▸<span data-testid="topbar__user__avatar" role="presentation" class="css-93m6oi">⋯</span>
        <p class="MuiTypography-root MuiTypography-body1" data-
        testid="topbar__user__name">demo"&gt;&lt;svg/onload=alert(1)&gt; nl&lt;img src=x onerror=alert(1)&gt;</p>
      </div>
     ▸<div class="css-ilnit7">⋯</div>
     ▸<div class="css-ilnit7">⋯</div>
     ▸<div class="css-ilnit7">⋯</div>
    </div>
   </div>
  </div>
  <div tabindex="0" data-test="sentinelEnd"></div>
 </div>
</body>
</html>
```

... Paper-rounded   div.MuiPaper-root.MuiPaper-elevation1.css-v2kfba.MuiPaper-rounded   div.css-ilnit7   p.MuiTypography-root.MuiTypography-body1 ...

The entity encoding was consistent across various reflected input fields:



https://nsecresilience-prod.thinklab.cloud/indicium/

think wise.

Indicium is running

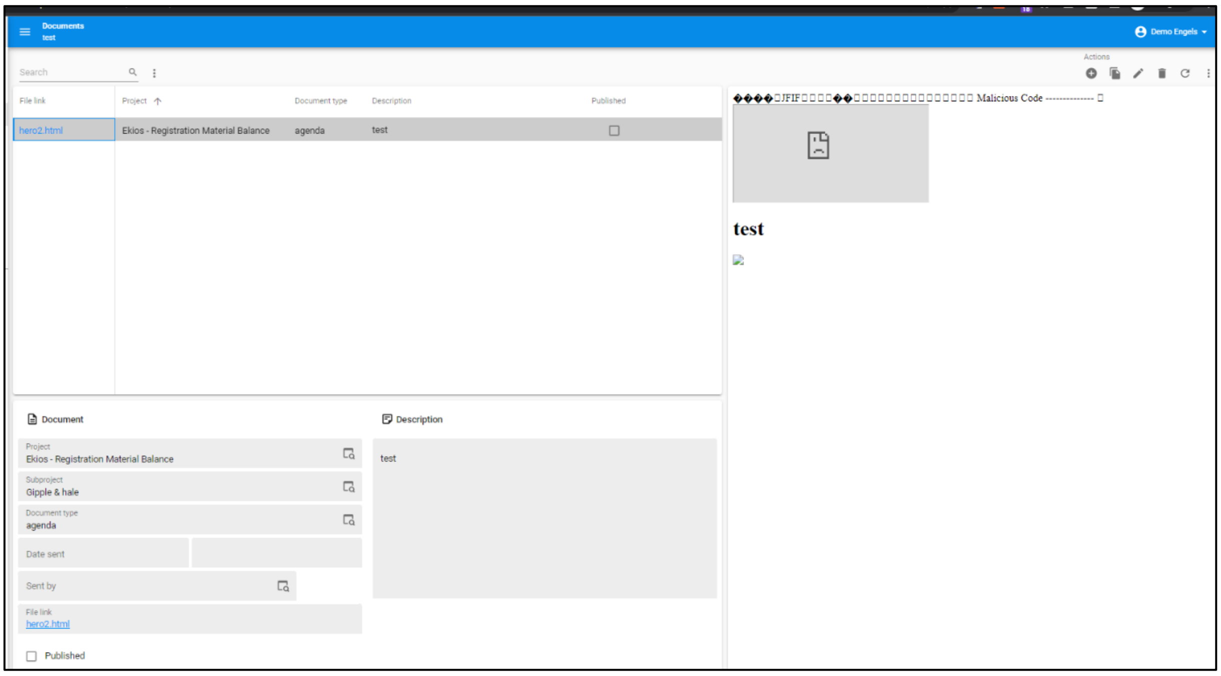
Version: 2023.1
Build: 10.0
Metasource version: 2023.1
Agent: 028ca2ed-eb20-4fa5-9edb-186e13140207

You are logged in as: demo"><svg/onload=alert(1)> nl<img src=x onerror=alert(1)>.

```
nkwise Software">
<h1 class="Title">Indicium is running</h1>
<br>
<strong>Version</strong>
": 2023.1 "
<br>
<strong>Build</strong>
": 10.0 "
<br>
<strong>Metasource version</strong>
<text>: 2023.1</text>
<br>
<strong>Agent</strong>
": 028ca2ed-eb20-4fa5-9edb-186e13140207 "
▼<p>
  "You are logged in as: "
  <strong class="Username" title="Username:
  demo_nl">demo"&gt;&lt;svg/onload=alert(1)&gt;
  nl&lt;img src=x onerror=alert(1)&gt;</strong>
  "."
</p>
▶<a class="Button DefaultButton" href="/indicium/account
ui/logout">⋯</a>
</div>
</div>
<br>
</body>
</html>
```

However some exceptions were found, mainly by making use of the earlier found unrestricted file upload vulnerabilities across file upload functions.

For example in the insights application -> projects -> documents section, it was possible to upload any file, which could hen be previewed inside the application. By uploading a .html file it was possible to get HTML code reflected inside the application:



Documents
test

Demo Engels

File link	Project	Document type	Description	Published
hero2.html	Ekios - Registration Material Balance	agenda	test	☐

����JFIF����������������������� Malicious Code --------------

test

Document

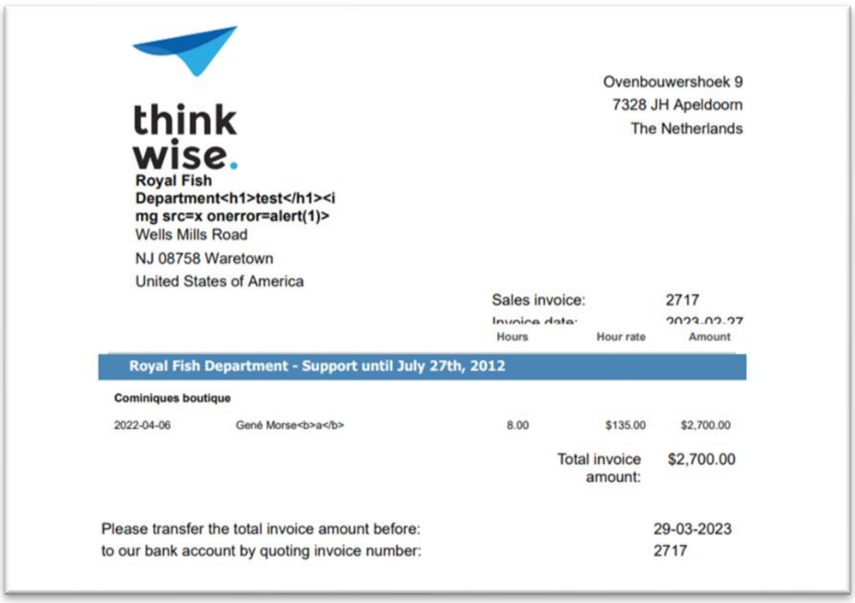
Project: Ekios - Registration Material Balance
Subproject: Gipple & hale
Document type: agenda
Date sent:
Sent by:
File link: hero2.html
☐ Published

Description

test

However XSS was still prevented in this context by the usage of the content-security-policy security header which is good. Nevertheless, HTML injection can be used to alter the appearance of a page for other users that visit it. Therefore uploaded files can be used for the purpose of social engineering attacks. Although it is good to be aware of this, no finding was included because HTML extensions can be blocked in the Software Factory.

As part of XSS-like injection attacks a PDF generation function was tested under finance -> generate invoice -> print invoice. The underlying reason is that server-side PDF generators are sometimes vulnerable for HTML injection or SSRF attacks. By changing user input parameters to XSS and SSRF related payloads it was checked whether the PDF generator would handle these payloads correctly. From the tests performed there were no signs of reflected HTML/XSS or SSRF-related callbacks.



think wise.

Ovenbouwershoek 9
7328 JH Apeldoorn
The Netherlands

Royal Fish Department<h1>test</h1><img src=x onerror=alert(1)>
Wells Mills Road
NJ 08758 Waretown
United States of America

Sales invoice: 2717
Invoice date: 2023-02-27

		Hours	Hour rate	Amount
Royal Fish Department - Support until July 27th, 2012				
Cominiques boutique				
2022-04-06	Gené Morse<b>a</b>	8.00	$135.00	$2,700.00
			Total invoice amount:	$2,700.00

Please transfer the total invoice amount before: 29-03-2023
to our bank account by quoting invoice number: 2717

No findings were added.

5.6 – SQL injection

With SQL injection, like XSS, input provided by a user is not checked sufficiently for malicious content. With SQL injection this vulnerability is used to influence SQL statements used in the web application to extract or manipulate information from the web application database.

The application and API were explored through OWASP ZAP, which supports recognition of OData functions, and subsequently tested for SQL injection.

No vulnerabilities were identified.

5.7 – SSL/TLS checks

The implementation of the encryption of internet traffic between the Insights demo application (nsecresilience.thinklab.cloud) and its users was analysed, with the following results:

- Certificate is delivered by Sectigo, and is assigned to *.thinklab.cloud
- The certificate is trusted and has a good validity (19th May, 2023)
- The webserver supports TLS 1.2, TLS 1.1 and TLS 1.0
- The cipher suites offered by the webserver to start session encryption are mostly adequate, but as a general rule it is advised to remove the TLS_RSA cipher suites. In general, an up to date advice can be found in paragraph 2.3 on:
    - https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices
- It is worth noting that the cipher suite names above are in the IANA format. These can be mapped to OpenSSL format through for example the table on https://testssl.sh/openssl-iana.mapping.html
- Other checks indicate that the other properties of SSL/TLS are secure. For example secure renegotiation is supported. Downgrade attack prevention (TLS_FALLBACK_SCSV) is not active, but although activating it is an improvement, there is not sufficient risk involved to warrant a finding to be included

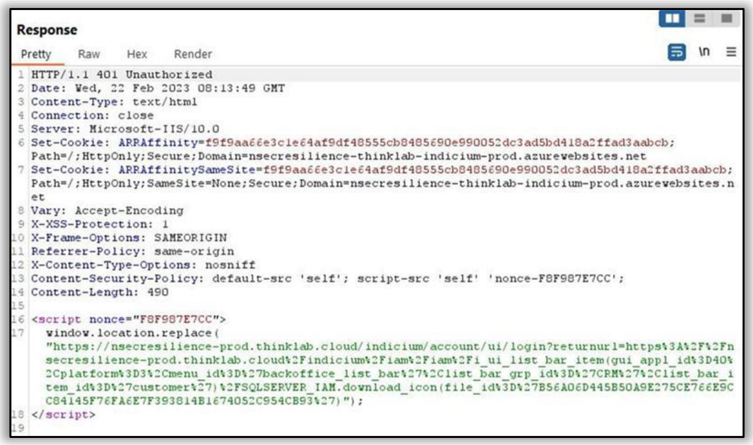
TLS 1.1 and TLS 1.0 are considered to be relatively weak protocols. The advice is to disable both. Because these findings are specific to the Insights application and not to the platform in general no finding will be included.

5.8 – Error handling

Analysis of error handling is important because in case exception- or errors are not processed correctly, sensitive information can be exposed to an attacker, for example a stack trace with directory names or filenames. The following conclusions as result of analysis:

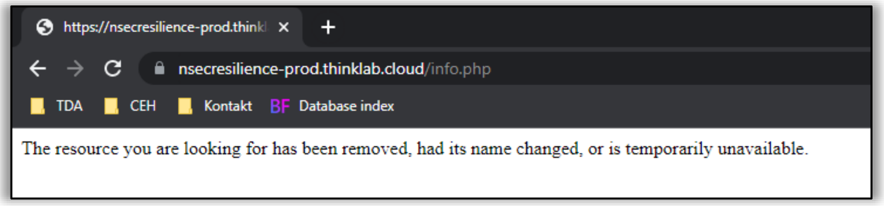
5.8.1 – Client errors (4xx)

- **401 Unauthorized:** If a page is accessed or a request is made to which a user should not have access, a 401 Unauthorized error is consistently returned with in the response that the user does not have access to the requested resource. This also applies to unauthenticated requests:



```
HTTP/1.1 401 Unauthorized
Date: Wed, 22 Feb 2023 08:13:49 GMT
Content-Type: text/html
Connection: close
Server: Microsoft-IIS/10.0
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.n
et
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self'; script-src 'self' 'nonce-F8F987E7CC';
Content-Length: 490

<script nonce="F8F987E7CC">
  window.location.replace(
  "https://nsecresilience-prod.thinklab.cloud/indicium/account/ui/login?returnurl=https%3A%2F%2Fn
  secresilience-prod.thinklab.cloud%2Findicium%2Fiam%2Fiam%2Fi_ui_list_bar_item(gui_appl_id%3D40%
  2Cplatform%3D3%2Cmenu_id%3D%27backoffice_list_bar%27%2Clist_bar_grp_id%3D%27CRM%27%2Clist_bar_i
  tem_id%3D%27customer%27)%2FSQLSERVER_IAM.download_icon(file_id%3D%27B56A06D445B50A9E275CE766E9C
  C84145F76FA6E7F393814B1674052C954CB93%27)");
</script>

```

- **404 Not found:** When requesting a non-existing page a standard IIS 404 error is shown. The formatting of a standard IIS error page is missing, but the text is exactly the same. Therefore, it can still be concluded that an IIS server is running.

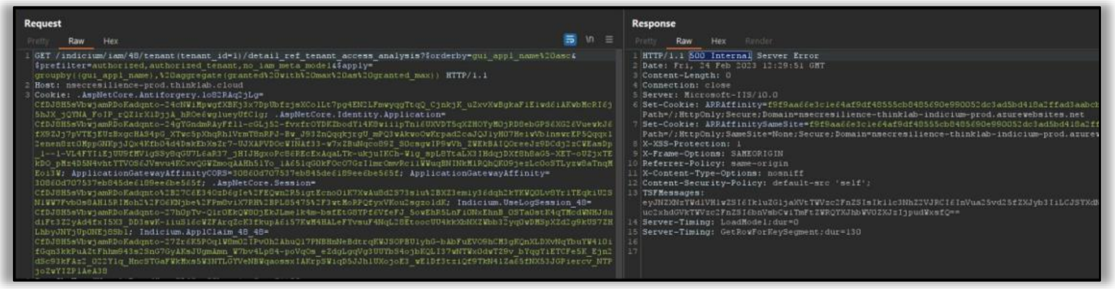

https://nsecresilience-prod.thinkl ×
nsecresilience-prod.thinklab.cloud/info.php
TDA
CEH
Kontakt
Database index

The resource you are looking for has been removed, had its name changed, or is temporarily unavailable.

Because these error messages do not reveal sensitive information, like verbose error responses, the handling of 4xx errors is adequate.

5.8.2 – Server errors (5xx)

- **500 Internal Server Error:** This message indicates that there is an error in your website's code. This is preventing the website from loading correctly. Using the Forgot Password feature, you have encountered a 500 internal server error:



```
GET /indicium/iam/48/tenant(tenant_id=1)/detail_ref_tenant_access_analysis?$orderby=gui_appl_name%20asc&
$prefilter=authorized,authorized_tenant,no_iam_meta_model&$apply=
groupby((gui_appl_name),%20aggregate(granted%20with%20max%20as%20granted_max)) HTTP/1.1
Host: nsecresilience-prod.thinklab.cloud
Cookie: .AspNetCore.Antiforgery.1o82RAq2jLg=
CfDJ8H5sVbwjamRDoKadqnto-24cNWiMpwgfXBKj3x7DpUbfzjsXColLt7pg4EN2LFmwyqgTtqQ_CjnkjK_uZxvXwBgkaFiEiwd6iAKwbMcRI6j
5hJX_jQYNA_FoIP_rQZirXiDjjA_hROe6wglueyUfClg; .AspNetCore.Identity.Application=
CfDJ8H5sVbwjamRDoKadqnto-24gYGndmRAyFf11-cGLj52-fvxfrOYDKZbodY14K8wiilpTn16UXVDT5qXZHOYyMOjRD8ebGPS6XG26VuewkJ6
fX9ZJj7pVTKjKUzBxgcHAS4pG_XTwc5pXhqRh1VrmT8nRPJ-Bw_J93ZnQqqkjrgU_mPQ3wAkwoOwKrpad2caJQJ1yH07HeiwVb1nswrKP5Qqqx1
2enen8ztOMppGNKpjJQx4Kfb04d4DmkEbXm2r7-UJXAPVDOcWINAf33-w7xZBuNqco89Z_S0cmgwIP9wVh_ZWKkBAIQOreeJz9DCdj2zCWEamDp
_1--1-VL4FYIiEjUU9fMVigSSy8qGU7L6aR37_jHIJHgxoPc86REcExAqaLTk-ukjuIKCh-Wig_mpL8YtaLX3IHdqjDXf8h8aG5-XET-oUZjxTE
kDO_pMz4D5N4vhtYTVOS6JVsvu4KCxvQGWZmoqAAHh51Yo_iA651qG0kFOcO7GzIlmrOmvRciIWWuqBNINkM1RQhQK09jezLcOoSTLyzw8aTnqM
Eoi3W; ApplicationGatewayAffinityCORS=30860d707537eb845de6189ee6be565f; ApplicationGatewayAffinity=
30860d707537eb845de6189ee6be565f; .AspNetCore.Session=
CfDJ8H5sVbwjamRDoKadqnto%2B27C6E340zD6gIe%2FEQwn2R5igtEcnoOiK7XwAu8d2S73slu%2BXZ3emly36dqh2kYKWQGLv8YriTEqkiU2S
N1WW7Fvb0s8AH15RIMoh2%2FO6KNjbe%2FPm8viX7RH%2BPL85475%2F3wtMoRPQfyxVKou2sgzoldK; Indicium.UseLogSession_48=
CfDJ8H5sVbwjamRDoKadqnto-27hOpTv-QlrOEkQW8OjEkJLme1k4m-bsfEtG8YPf6VfeFJ_5owEhR5LnFiONxEhnB_OSTa0stK4qTMcdWNHJdu
diFt3Z2yAd4fxI5X3_DD3swK-iiuS16cWZFArqZcK3fkupA6i57KwM4HALeFYvmuF4NqL28EtoocUU4kkXbNXZWbb3ZyqOwDMSpXZd2g9kU57ZH
LhbyJNYjUpONEj6Sbl; Indicium.ApplClaim_48_48=
CfDJ8H5sVbwjamRDoKadqnto-27Zr6K5POqlWBmO2IPvOh2AhuQ17PNBHnNeBdtrqKWJSOPBUlyhG-bAbFuEVO9hCM3gKQnXLDXvNqYbuYW410i
fGqn3kkPuAZtFhhm843s2SnG7GyAKsJUgmAmn_W7bv4Lp84-poVqOm_eZdgLgqVg3UUYbS4ojbKQLI37wNTWxOdwT29v_bYqgYiETCFe5K_Ejn2
dSc93kFAz2_O22Y1q_HncSTGaFWkMxs5WINTLGYVeNBWqaosmxIAKrpSWiqD5JJhlUXojoE3_wE1Df3tziQf9TkN41Za65fNX53JGPiercv_NTF
joZwYIZP1AeA38
```

```
HTTP/1.1 500 Internal Server Error
Date: Fri, 24 Feb 2023 12:29:51 GMT
Content-Length: 0
Connection: close
Server: Microsoft-IIS/10.0
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb
Path=/;HttpOnly;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
Set-Cookie: ARRAffinitySameSite=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ff
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurew
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self';
TSFMessages:
eyJNZXNzYWdlVHlwZSI6IkluZGljaXVtTWVzc2FnZSIsIklic3NhZ2VJRCI6InVua25vd25fZXJyb3IiLCJSYXdN
uc2xhdGVkTWVzc2FnZSI6bnVsbCwiTmFtZWRQYXJhbWVOZXJzIjpudWxsfQ==
Server-Timing: LoadModel;dur=0
Server-Timing: GetRowForKeySegment;dur=130
```

- Because these error messages do not reveal sensitive information, like verbose error responses, the handling of 5xx errors is adequate.

5.9 – Sensitive data exposure

With sensitive data exposure, information can be found which gives an attacker additional information with respect to the application or application landscape.

As per the OData standard, metadata was found, for example for the Insights application through the following URL:

- https://nsecresilience-prod.thinklab.cloud/indicium/iam/40/$metadata

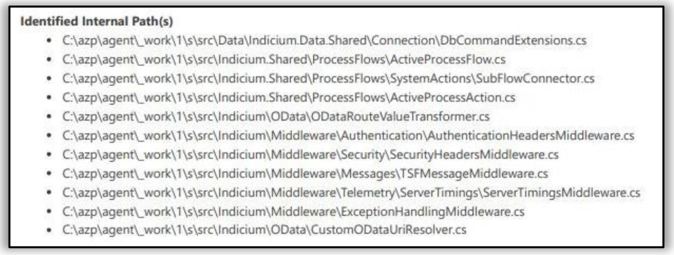
It is advised to hide the OData service metadata. The metadata describes the structure of the entities exposed by the service. Attackers can use the metadata document to better understand the structure of the entities exposed by the OData service and create more targeted attacks based on this information.

Since the metadata for Thinkwise applications can be disabled via a setting in appsettings.json no finding will be included.

Next to checking content of the JavaScript files and other HTTP traffic, brute force directory scanning was performed.

In addition, some possible internal file paths had been found. This was available from the following URL:

- https://nsecresilience-prod.thinklab.cloud/indicium/errorLog

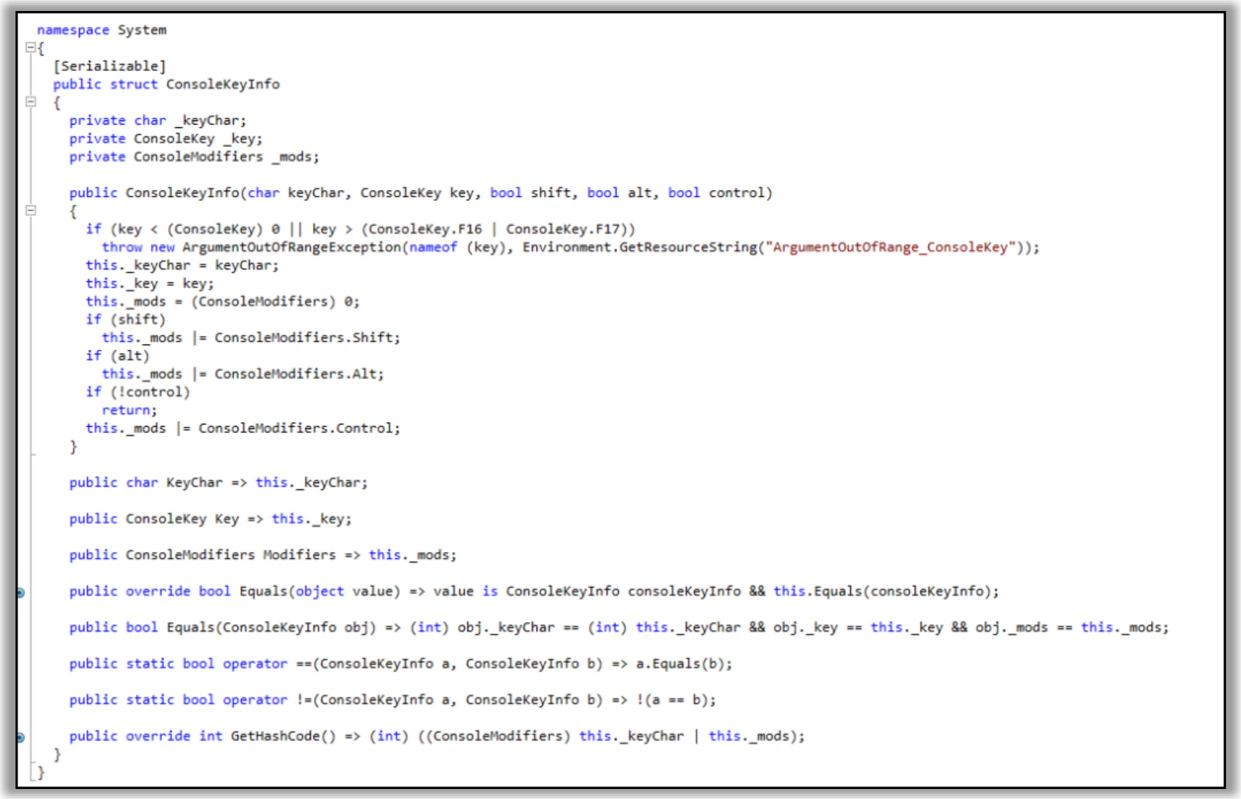


Identified Internal Path(s)

- C:\azp\agent_work\1\s\src\Data\Indicium.Data.Shared\Connection\DbCommandExtensions.cs
- C:\azp\agent_work\1\s\src\Indicium.Shared\ProcessFlows\ActiveProcessFlow.cs
- C:\azp\agent_work\1\s\src\Indicium.Shared\ProcessFlows\SystemActions\SubFlowConnector.cs
- C:\azp\agent_work\1\s\src\Indicium.Shared\ProcessFlows\ActiveProcessAction.cs
- C:\azp\agent_work\1\s\src\Indicium\OData\ODataRouteValueTransformer.cs
- C:\azp\agent_work\1\s\src\Indicium\Middleware\Authentication\AuthenticationHeadersMiddleware.cs
- C:\azp\agent_work\1\s\src\Indicium\Middleware\Security\SecurityHeadersMiddleware.cs
- C:\azp\agent_work\1\s\src\Indicium\Middleware\Messages\TSFMessageMiddleware.cs
- C:\azp\agent_work\1\s\src\Indicium\Middleware\Telemetry\ServerTimings\ServerTimingsMiddleware.cs
- C:\azp\agent_work\1\s\src\Indicium\Middleware\ExceptionHandlingMiddleware.cs
- C:\azp\agent_work\1\s\src\Indicium\OData\CustomODataUriResolver.cs

Further investigation revealed that this URL was only available to Administrators of the platform. Since this is only available to Administrators, no finding was raised for this.

Checks were also performed on Thinkwise's software factory application. These checks for example tried to validate whether the source code of this application could be inspected. This was indeed the case. Free .NET decompilers and code viewers such as DotPeek can be used to view the decompiled source without any limitations; no obfuscation is currently present within the source code:



```
namespace System
{
  [Serializable]
  public struct ConsoleKeyInfo
  {
    private char _keyChar;
    private ConsoleKey _key;
    private ConsoleModifiers _mods;

    public ConsoleKeyInfo(char keyChar, ConsoleKey key, bool shift, bool alt, bool control)
    {
      if (key < (ConsoleKey) 0 || key > (ConsoleKey.F16 | ConsoleKey.F17))
        throw new ArgumentOutOfRangeException(nameof (key), Environment.GetResourceString("ArgumentOutOfRange_ConsoleKey"));
      this._keyChar = keyChar;
      this._key = key;
      this._mods = (ConsoleModifiers) 0;
      if (shift)
        this._mods |= ConsoleModifiers.Shift;
      if (alt)
        this._mods |= ConsoleModifiers.Alt;
      if (!control)
        return;
      this._mods |= ConsoleModifiers.Control;
    }

    public char KeyChar => this._keyChar;

    public ConsoleKey Key => this._key;

    public ConsoleModifiers Modifiers => this._mods;

    public override bool Equals(object value) => value is ConsoleKeyInfo consoleKeyInfo && this.Equals(consoleKeyInfo);

    public bool Equals(ConsoleKeyInfo obj) => (int) obj._keyChar == (int) this._keyChar && obj._key == this._key && obj._mods == this._mods;

    public static bool operator ==(ConsoleKeyInfo a, ConsoleKeyInfo b) => a.Equals(b);

    public static bool operator !=(ConsoleKeyInfo a, ConsoleKeyInfo b) => !(a == b);

    public override int GetHashCode() => (int) ((ConsoleModifiers) this._keyChar | this._mods);
  }
}
```

It is advised to obfuscate the source code for the Software Factory. This can help protect the intellectual property of software companies or individual developers. A finding with low severity has been raised for this.

5.10 – Security (mis)configuration

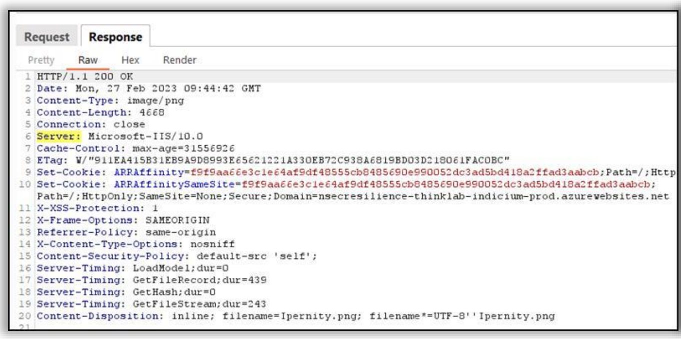
For adequate security it is necessary that the correct configuration is chosen and implemented for all parts. This applies to the application but also to the webserver. Several checks were performed, whose main results are listed here.

5.10.1 – Stack information in HTTP response messages

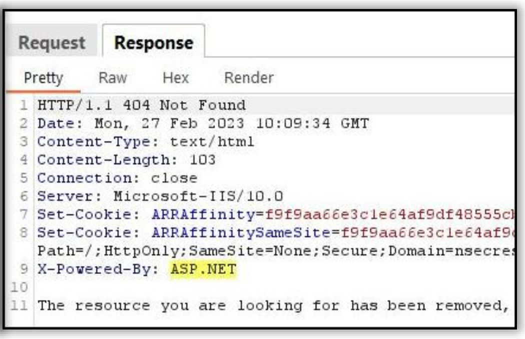
An attacker will start a possible attack by making an overview of the application stack. The HTTP response message is an important instrument for gathering information about the application stack. It is considered best practice to remove as much information from the HTTP response messages as possible.

- Server: Microsoft-IIS/10.0
- X-Powered-By: ASP.NET

It is advised to remove these response headers if possible. This finding is specific for the Insights application and does not apply to the Thinkwise platform in general.



```
HTTP/1.1 200 OK
Date: Mon, 27 Feb 2023 09:44:42 GMT
Content-Type: image/png
Content-Length: 4668
Connection: close
Server: Microsoft-IIS/10.0
Cache-Control: max-age=31556926
ETag: W/"911EA415B31EB9A9D8993E65621221A330EB72C938A6819BD03D218061FAC0BC"
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;Path=/;Http
Set-Cookie: ARRAffinitySameSite=f9f9aa66e3c1e64af9df48555cb8485690e990052dc3ad5bd418a2ffad3aabcb;
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecresilience-thinklab-indicium-prod.azurewebsites.net
X-XSS-Protection: 1
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self';
Server-Timing: LoadModel;dur=0
Server-Timing: GetFileRecord;dur=439
Server-Timing: GetHash;dur=0
Server-Timing: GetFileStream;dur=243
Content-Disposition: inline; filename=Ipernity.png; filename*=UTF-8''Ipernity.png
```



```
HTTP/1.1 404 Not Found
Date: Mon, 27 Feb 2023 10:09:34 GMT
Content-Type: text/html
Content-Length: 103
Connection: close
Server: Microsoft-IIS/10.0
Set-Cookie: ARRAffinity=f9f9aa66e3c1e64af9df48555c
Set-Cookie: ARRAffinitySameSite=f9f9aa66e3c1e64af9
Path=/;HttpOnly;SameSite=None;Secure;Domain=nsecre
X-Powered-By: ASP.NET

The resource you are looking for has been removed,
```

5.10.2 – HTTP Security Headers

Security headers are directives used by web applications to configure security defenses in web browsers. Based on these directives, browsers can make it harder to exploit client-side vulnerabilities such as Cross-Site Scripting or Clickjacking.

Headers that are present:

- Present headers:
- X-XSS-Protection
- CSP header
- Referrer policy
- X-Frame-Options
- X-Content-Type-Options

Missing headers:

- Strict-Transport-Security

With the Strict-Transport-Security security header, you ensure that users can only access the Web application via the HTTPS protocol and that any future attempts to access it using HTTP should automatically be converted to HTTPS. It is advised to always add this header. However because this is usually an infrastructure level setting, this header will not be set from the Thinkwise platform and no finding will therefore be incorporated.

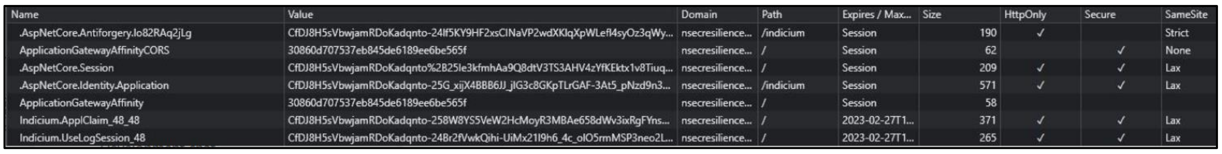
5.10.3 – Cookie settings

Cookies used by the applications to store important data on the user's side. It is important to protect this data. It can be done in part by providing the correct settings to the cookies from the server-side. On the one hand, it is good to give important cookies the so-called Secure flag, which forces these cookies only to be sent over secure connections (HTTPS). To prevent cookies from being vulnerable to cross-site scripting (XSS attacks), cookies can be provided with the flag HttpOnly, which prevents a cookie from being read by an attacker. Another best practice is the SameSite cookie attribute that helps in preventing CSRF.

The .AspNetCore.Identity.Application is used for access control and typically security settings should be evaluated for that cookie.

For this specific cookie, the settings that are available for cookies that will avoid cookie values being intercepted through a man-in-the-middle attack (“secure”) and through XSS (“httponly”) are both active.

The “Samesite” setting however is currently set to “Lax” which means that a number of CSRF scenario’s are not blocked. It is advised to change this setting to “Strict” if possible. However because an antiforgery cookie is also being used, which should mitigate CSRF attacks as well, no findings are included.



Name	Value	Domain	Path	Expires / Max...	Size	HttpOnly	Secure	SameSite
.AspNetCore.Antiforgery.Io82RAq2jLg	CfDJ8H5sVbwjamRDoKadqnto-24If5KY9HF2xsCINaVP2wdXKIqXpWLefl4syOz3qWy...	nsecresilience...	/indicium	Session	190	✓		Strict
ApplicationGatewayAffinityCORS	30860d707537eb845de6189ee6be565f	nsecresilience...	/	Session	62		✓	None
.AspNetCore.Session	CfDJ8H5sVbwjamRDoKadqnto%2B25Ie3kfmhAa9Q8dtV3TS3AHV4zYfKEktx1v8Tiuq...	nsecresilience...	/	Session	209	✓	✓	Lax
.AspNetCore.Identity.Application	CfDJ8H5sVbwjamRDoKadqnto-25G_xijX4BBB6JJ_jIG3c8GKpTLrGAF-3At5_pNzd9n3...	nsecresilience...	/indicium	Session	571	✓	✓	Lax
ApplicationGatewayAffinity	30860d707537eb845de6189ee6be565f	nsecresilience...	/	Session	58			
Indicium.ApplClaim_48_48	CfDJ8H5sVbwjamRDoKadqnto-258W8YS5VeW2HcMoyR3MBAe658dWv3ixRgFYns...	nsecresilience...	/	2023-02-27T1...	371	✓	✓	Lax
Indicium.UseLogSession_48	CfDJ8H5sVbwjamRDoKadqnto-24Br2fVwkQihi-UiMx21I9h6_4c_olO5rmMSP3neo2L...	nsecresilience...	/	2023-02-27T1...	265	✓	✓	Lax

6 – Audit results

Because software created using recent Microsoft technology is in obvious areas often secure by default, the quality and coverage of security testing can be raised by performing further audit like checks on the software platform.

These white box checks are proposed to be performed on location, where an nSEC/Resilience consultant will perform the checks together with Thinkwise resources. For the audit activities in performed in 2023 focus was placed on the most relevant attention points as identified during a previous audit in 2020. Backend code security was investigated using sample based code inspection with developers on processing of user input that is processed in the backend, for example in TSQL. In addition to this a security analysis was performed of the overall application architecture with the purpose of identifying potential security issues or weaknesses in interfaces or application entry/exit points.

6.1 – General comments

Newer versions of the Thinkwise platform components Indicium and Frontend have been made available recently. The older versions of these components can be considered to be effectively end of life and have been placed out of scope for the audit.

Thinkwise currently already performs a number of security related checks as part of their CI/CD pipeline:

- Static analysis based on Sonarcloud and Eslint for frontend/Universal
- Open source component checks frontend/Universal: Yarn

Open source component checks for other components are performed on ad hoc basis and are performed manually (NuGET repositories).

6.2 – (T)SQL for business logic

The functionality of an application made with the Thinkwise platform is generated based on what is defined in the Software Factory and the Indicium middle layer. This process also includes generating parameterized (T)SQL statements

- Default procedures that are used for example to work with input forms on application level
- Layout procedures that are meant to indicate which fields should be visible and which not
- Context procedures that process tasks or generate reports
- Database triggers
- Batch procedures
- Process procedures, through which workflows are defined

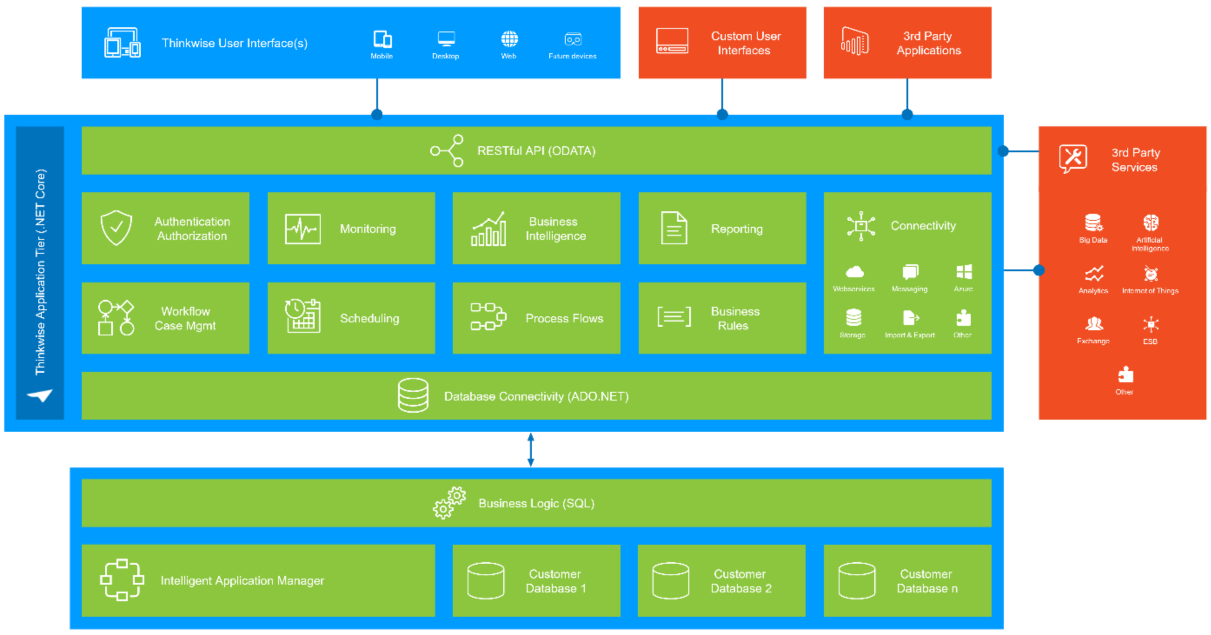
(T)SQL statements can only be edited by users that have access to the database or Software Factory. These types of access are typically not obtained by users of the end client but only by Thinkwise or partner organizations.

Because the (T)SQL statements are generated at runtime, they can't be checked using static code security checkers such as sonarcube.

Based on what was discussed in relation to the (T)SQL statements, no risks were identified because all generated statements are parameterized and there is no opportunity to change these statements.

6.3 – Implementation database connection (ADO.NET)

In the application landscape of an active Thinkwise application at runtime, the Indicium middle layer has an active connection to the IAM component based on connection settings that are set in the platform by default.



The configuration for any connections that need to be made to other databases are stored in the IAM. The configurations are made in the Software Factory, which then synchronizes to the IAM.

The passwords for the connections were also stored without encryption for the Insights application. However for the Thinkwise platform in general it is possible to store these passwords encrypted.

Another observation was related to the way the initial credentials for the database pool are stored. Also due to the currently available documentation (docs.thinkwisesoftware.com/docs/deployment/indicium) these credentials are often stored hardcoded in appsettings.json. More secure alternatives are available; therefore no finding will be included.

6.4 – Preview components

Based on a discussion of the application landscape, a potential attack vector was found in the availability of file preview functionality in the platform. For example, HTML files can be viewed as PDF, and Excel files can be viewed to HTML files.

The preview components make use of the gembox library (gemboxsoftware.com). Thinkwise has obtained a license to make use of gembox as a part of the Thinkwise platform.

Based on the audit discussions, a number of additional tests have been defined for the penetration test. No findings have been done for the preview functionality itself. The gembox software does not seem to contain any known vulnerabilities or default configuration errors.

An observation that was done is the fact that the gembox licence key is included hardcoded in the Thinkwise platform files. However this key is not usable from other environments because it is digitally signed.

7 – Conclusion and recommendations

During the initial penetration test on the example application of the Thinkwise platform no findings of high or critical severity were done. Testers did not succeed in obtaining significant amounts of sensitive data nor take control of the server. This is a good result.

The audit also did not produce findings that required immediate actions.

After receiving feedback from Thinkwise on the initial findings, and retesting of fixed findings, only a small number of low severity findings remained. These findings of low severity can be seen as findings for which there is no direct urgency to address them, but implementing solutions for these findings will further raise the bar for potential attackers and establish a baseline that would give customers or external auditors increased confidence.

A findings overview with the vulnerabilities ordered by severity can be found on the next page.

Description	Category	Severity	Advice
In some cases sessions remain active after long periods of inactivity	Authentication & Session Management	Low	Enforce that the mechanism to automatically end sessions based on user inactivity works in all situations
No brute force protection on the MFA token for the password reset function	Authentication & Session Management	Low	Add functionality to block a user or a code after 5 incorrect attempts within a short timeframe
Runtime components of the Thinkwise platform can be decompiled into readable source code	Sensitive Data Exposure	Low	It is advised to obfuscate/hide the source code to protect intellectual property

Appendix A – Results DNS reconnaissance

Subdomain	IP address
community.thinkwisesoftware.com	52.222.139.88
docs.thinkwisesoftware.com	144.178.66.249
office.thinkwisesoftware.com	144.178.66.249
updates.thinkwisesoftware.com	144.178.66.249
ssh.thinkwisesoftware.com	185.162.30.162
universal.thinkwisesoftware.com	144.178.66.244
msoid.thinkwisesoftware.com	40.126.32.68
lyncdiscover.thinkwisesoftware.com	52.112.196.12
www.thinkwisesoftware.com	199.60.103.28
metrics.thinkwisesoftware.com	144.178.66.249
staging.thinkwisesoftware.com	199.60.103.228
mail.thinkwisesoftware.com	144.178.66.243
offers.thinkwisesoftware.com	199.60.103.28
enterpriseregistration.thinkwisesoftware.com	20.190.137.40
registry.thinkwisesoftware.com	195.154.68.114
autodiscover.thinkwisesoftware.com	144.178.66.243
kibo.thinkwisesoftware.com	144.178.66.243
vpn.thinkwisesoftware.com	144.178.66.242
blog.thinkwisesoftware.com	199.60.103.28
sip.thinkwisesoftware.com	52.112.192.11
enterpriseenrollment.thinkwisesoftware.com	20.91.147.72
insights.thinkwisesoftware.com	144.178.66.244
filecap.thinkwisesoftware.com	144.178.66.245
thinklab.thinkwisesoftware.com	185.149.37.40
prtg.thinkwisesoftware.com	144.178.66.244
tsf-quarantaine.thinkwisesoftware.com	144.178.66.242
webmail.thinkwisesoftware.com	95.97.179.163
licensing.thinkwisesoftware.com	144.178.66.249