Open-Xchange Security Advisory 2021-07-15

From: Martin Heiland via Fulldisclosure <fulldisclosure () seclists org>
Date: Thu, 15 Jul 2021 14:09:28 +0200

```
Dear subscribers,

we're sharing our latest advisory with you and like to thank everyone who contributed in finding and solving those
vulnerabilities. Feel free to join our bug bounty programs for OX AppSuite, Dovecot and PowerDNS at HackerOne.

Note that some bugfixes (MWB-423, MWB-460, MWB-492, MWB-493 and MWB-494) have been fixed with 7.10.4 and later
already.
We recently provided backports to 7.10.3, thus updating the information on those vulnerabilities.

Yours sincerely,
Martin Heiland, Open-Xchange GmbH


Product: OX App Suite, OX Guard, OX Documents
Vendor: OX Software GmbH



Internal reference: MWB-423
Vulnerability type: Server-Side Request Forgery (CWE-918)
Vulnerable version: 7.10.3
Vulnerable component: backend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.3-rev32
Vendor notification: 2020-06-26
Solution date: 2021-02-10
Public disclosure: 2021-07-15
Researcher Credits: Stuart Redman
CVE reference: CVE-2020-24700
CVSS: 6.4 (CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:N)

Vulnerability Details:
The oAuth Proxy capability, used to exchange data with third-party services such as Twitter, can be abused to craft
requests to services which are prohibited. These services may reside within a protected network and could be exposed
using this technique. The code to check for allowed domains did not account for certain URL constructs.

Risk:
Malicious users can trigger network requests to web services outside of the expected trust boundary, for example
services within a restricted network to which the OX App Suite middleware node has access. In case such services do
not
have further access control, a malicious user could retrieve web service content from them. The vulnerability allows
to
control request type and headers sent to those services.

Steps to reproduce:
1. Connect your OX App Suite account to an oAuth-enabled service like Twitter
2. Forge API requests via /api/oauth/proxy containing payload related to internal services
3. API response will contain an error but also the retrieved content for the internal service

Proof of concept:
PUT https://example.com/appsuite/api/oauth/proxy?api=com.openexchange.oauth.twitter&session=XYZ
{"url":"https://twitter.com () internal example com","params":{"count":10,"include_entities":true}}

Solution:
We improved detection of user-provided payload when checking against access lists. Regardless of this fix we suggest
tight network segmentation, egress traffic filtering and access controls for any kind of service.



---



Internal reference: MWB-460
Vulnerability type: Server-Side Request Forgery (CWE-918)
Vulnerable version: 7.10.3
Vulnerable component: backend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.3-rev32
Vendor notification: 2020-07-07
Solution date: 2021-02-10
Public disclosure: 2021-07-15
CVE reference: CVE-2020-24700
CVSS: 4.3 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N)

Vulnerability Details:
External mail account discovery allows malicious users to append arbitrary URL paths to mail addresses. In combination
with malicious auto-configuration DNS records, this can be abused to access web services outside of the expected trust
boundary, regardless of existing blocklists.

Risk:
Malicious users can trigger network requests to web services outside of the expected trust boundary, regardless of
existing blocklists. This may be used to probe for services and paths within a restricted network to which the OX App
Suite middleware node has access and potentially ease further attacks.

Steps to reproduce:
1. Setup a DNS A record for autoconfig.example.com, pointing to a local addresses like 127.0.0.1
2. Use the "external mail account" feature to setup a mail account for this domain
3. Append URL paths to the mail address, e.g. foo () example com/ssrf/ping

Proof of concept:
DNS lookup will return "127.0.0.1" and OX App Suite will append the URL fragment of the mail address, resulting in a
GET request to http://127.0.0.1/ssrf/ping?emailaddress=foo () example com.

Solution:
We restricted the ability to access blocked networks when performing autoconfig lookups.



---



Internal reference: MWB-492
Vulnerability type: Cross-Site Scripting (CWE-80)
Vulnerable version: 7.10.3
Vulnerable component: backend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.3-rev32
Vendor notification: 2020-07-20
```

Solution date: 2021-02-10
Public disclosure: 2021-07-15
CVE reference: CVE-2020-24701
CVSS: 4.3 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N)

Vulnerability Details:
The "debug" option for the /apps/manifests endpoint included request parameters in its response, without using HTML escaping.

Risk:
Malicious script code can be executed within a users context. This can lead to session hijacking or triggering unwanted
actions via the web interface (e.g. redirecting to a third-party site). To exploit this an attacker would require the victim to follow a hyperlink.

Steps to reproduce:
1. Create a link to the /apps/manifest endpoint using the debug option and append malicious script code
2. Make a user open this link, for example through social engineering

Proof of concept:
https://example.com/ajax/apps/manifests?action=all&format=debug&xss=%3Cscript%3Ealert(%22XSS%22);%3C/script%3E

Solution:
We now escape any user-provided content when creating the debug response.

Internal reference: MWB-493
Vulnerability type: Cross-Site Scripting (CWE-80)
Vulnerable version: 7.10.4 and earlier
Vulnerable component: backend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.3-rev32
Vendor notification: 2020-07-20
Solution date: 2021-02-10
Public disclosure: 2021-07-15
CVE reference: CVE-2020-24701
CVSS: 4.3 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N)

Vulnerability Details:
The logic for determining safe content could be bypassed by providing unknown values for content-disposition while requesting a shared file. In case the file contained malicious script code, this would be executed.

Risk:
Malicious script code can be executed within a users context. This can lead to session hijacking or triggering unwanted
actions via the web interface (e.g. redirecting to a third-party site). To exploit this an attacker would require the victim to follow a hyperlink.

Steps to reproduce:
1. Create a HTML file with malicious JS code and upload it to Drive
2. Create a public sharing link
3. Modify this link to contain a unexpected content_disposition parameter value
4. Make the victim follow this link

Proof of concept:
https://example.com/ajax/share/<share-token>?delivery=view&content_disposition=foo

Solution:
We improved the detection mechanism to neglect user-specified parameter values.

Internal reference: MWB-494
Vulnerability type: Cross-Site Scripting (CWE-80)
Vulnerable version: 7.10.3
Vulnerable component: backend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.3-rev32
Vendor notification: 2020-07-21
Solution date: 2021-02-10
Public disclosure: 2021-07-15
CVE reference: CVE-2020-24701
CVSS: 4.3 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N)

Vulnerability Details:
Access to a cache for internal file handling (e.g. importing vcards from an E-Mail to the address book) was not restricted to specific users. While the chance of unauthorized access is very low, the attacker would have required to correctly guess a 128b UUID before the cache expires, this could be used to hide and deliver malicious script code. Content at this cache was not sanitized or filtered and direct references could be used in phishing attacks.

Risk:
Malicious script code can be executed within a users context. This can lead to session hijacking or triggering unwanted
actions via the web interface (e.g. redirecting to a third-party site). To exploit this an attacker would require the victim to follow a hyperlink.

Steps to reproduce:
1. Include malicious script code within external content like a vcard file
2. Attach this file to a mail and use the conversion API to create a managed distributed file
3. Find out the UUID reference to this managed "distributedFile"
4. Make the victim open this direct reference as hyperlink

Solution:
We now require user-specific authentication to access the API endpoint for managed distributed files.

Internal reference: MWB-838
Vulnerability type: Cross-Site Scripting (CWE-80)
Vulnerable version: 7.10.4 and earlier
Vulnerable component: backend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.3-rev32, 7.10.4-rev18
Vendor notification: 2021-01-04
Solution date: 2021-01-11
Public disclosure: 2021-07-15
CVE reference: CVE-2021-26698
CVSS: 3.5 (CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:U/C:L/I:N/A:N)

Vulnerability Details:
In case a legacy component ("dataretrieval", disabled by default) is installed and enabled, it can be exploited to serve script code that can be called by a direct reference. The component did lack proper sanitization and output filtering.

Risk:
Malicious script code can be executed within a users context. This can lead to session hijacking or triggering unwanted
actions via the web interface (e.g. redirecting to a third-party site). To exploit this an attacker would require the victim to follow a hyperlink.

Steps to reproduce:
1. As Operator, enable the "dataretrieval" component
2. As attacker, upload script-code as binary data
3. Distribute a direct reference to the dataretrieval endpoint to the victim

4. Make the victim open this direct reference as hyperlink

Solution:
We removed the legacy feature to avoid unintended usage. Note that this does NOT affect any GDPR related data export functionality.

Internal reference: MWB-839
Vulnerability type: Cross-Site Scripting (CWE-80)
Vulnerable version: 7.10.4 and earlier
Vulnerable component: backend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.3-rev32, 7.10.4-rev18
Vendor notification: 2021-01-04
Solution date: 2021-02-10
Public disclosure: 2021-07-15
CVE reference: CVE-2021-26698
CVSS: 5.3 (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)

Vulnerability Details:
A URL parameter could be used to modify the result of existing sanitization and output handling, when downloading user-generated content.

Risk:
Malicious script code can be executed within a users context. This can lead to session hijacking or triggering unwanted
actions via the web interface (e.g. redirecting to a third-party site). To exploit this an attacker would require the victim to follow a hyperlink.

Steps to reproduce:
1. As attacker, upload a code snippet to drive and create a sharing link
2. Modify the URL parameters to include the "dl" parameter
3. Embed a direct reference to this snippet at a malicious website or make a user follow the reference

Solution:
We now ignore user-provided URL parameters when deciding how to handle output. References to shared files will always trigger downloads.

Internal reference: OXUIB-645
Vulnerability type: Cross-Site Scripting (CWE-80)
Vulnerable version: 7.10.4 and earlier
Vulnerable component: frontend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.3-rev27, 7.10.4-rev19
Vendor notification: 2021-01-04
Solution date: 2021-02-10
Public disclosure: 2021-07-15
CVE reference: CVE-2021-26698
CVSS: 4.3 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N)

Vulnerability Details:
The "app loader" mechanism of the frontend component could be abused to load content from relative URLs, outside of the
intended code loading API path. This can be used by attackers to add references to malicious content that is served by the same domain.

Risk:
Malicious script code can be executed within a users context. This can lead to session hijacking or triggering unwanted
actions via the web interface (e.g. redirecting to a third-party site). To exploit this an attacker would require the victim to follow a hyperlink.

Steps to reproduce:
1. As attacker, upload a code snippet to drive and create a sharing link
2. Modify the "app loader" URL and include a relative reference to the shared code snippet
3. Embed a direct reference to this snippet at a malicious website or make a user follow the reference

Solution:
We now restrict relative references to only include the intended API path.

Internal reference: DOCS-3139
Vulnerability type: Server-Side Request Forgery (CWE-918)
Vulnerable version: 7.10.4 and earlier
Vulnerable component: imageconverter
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.3-rev4, 7.10.4-rev4
Vendor notification: 2020-12-18
Solution date: 2021-02-10
Public disclosure: 2021-07-15
CVE reference: CVE-2021-26699
CVSS: 5.4 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:L)

Vulnerability Details:
SVG files are processed by the imageconverter component. In case they include references to external entities, imageconverter would attempt to process them.

Risk:
This technique can be used to reduce availability of the environment by referencing excessive amounts of data. It may also be used to track individual users and monitor what files they are opening using App Suite. This would require the attacker to inject compromised images to the users workflow.

Steps to reproduce:
1. Generate a SVG file with external references, e.g. API endpoints
2. Rename the file to add a .png extension and share it on OX Drive or send by mail
3. Make the victim use the image viewer to open the file

Solution:
We now restrict relative references and block potentially harmful files from being processed as images.

Attachment: signature.asc
Description: Message signed with OpenPGP

By Date By Thread

Current thread:

Open-Xchange Security Advisory 2021-07-15 ***Martin Heiland via Fulldisclosure (Jul 16)***