Open-Xchange Security Advisory 2020-10-13

From: Open-Xchange GmbH via Fulldisclosure <fulldisclosure () seclists org>
Date: Tue, 13 Oct 2020 09:09:21 +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 App Suite, Dovecot and PowerDNS at HackerOne.

Yours sincerely,
Martin Heiland, Open-Xchange GmbH



Product: OX App Suite / OX Documents
Vendor: OX Software GmbH



Vulnerability type: Cross-Site Scripting (CWE-80)
Vulnerable version: 7.10.2, 7.10.3
Vulnerable component: backend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.2-rev29, 7.10.3-rev15
Vendor notification: 2020-04-27
Solution date: 2020-07-01
Public disclosure: 2020-10-13
Researcher Credits: MOGWAI LABS GmbH
CVE reference: CVE-2020-15004
CVSS: 2.2 (CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:L/I:N/A:N)

Vulnerability Details:
A internal diagnostics servlet did return the content of a HTTP GET request as part of the generated website. This can
be used to supply malicious JS code via a hyperlink. Access to the servlet is unauthenticated and not possible over a
public network by default.

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).

Steps to reproduce:
1. Create a link to the diagnostics servlet containing script code
2. Make someone with access to this servlet click the link

Proof of concept:
http://example.com:8009/stats/diagnostic?param=%3Cscript%3Ealert(%27ayb%27);%3C/script%3E%22

Solution:
We no longer return any supplied parameter as part of the HTML page.



---



Internal reference: MWB-289
Vulnerability type: Information exposure (CWE-200)
Vulnerable version: 7.10.2, 7.10.3
Vulnerable component: backend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.2-rev29, 7.10.3-rev15
Vendor notification: 2020-05-08
Solution date: 2020-07-01
Public disclosure: 2020-10-13
CVE reference: CVE-2020-15003
CVSS: 3.7 (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)

Vulnerability Details:
When accessing a public or restricted share as a guest, for example in Drive, users have the ability to query and
terminate sessions of other guests. This exposes IP addresses, os and user agent information as well as session
identifiers.

Risk:
Malicious guest users are able to terminate other users sessions. They can also look up other users IP addresses and
client information.

Steps to reproduce:
1. Create a shared Drive folder
2. Have several guests visit this share
3. As a guest, query the session API, check Settings -> Security

Solution:
We removed the ability for guests to access session information of other guests.



---



Internal reference: MWB-348
Vulnerability type: Server-side request forgery (CWE-918)
Vulnerable version: 7.10.3 and earlier
Vulnerable component: backend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.8.4-rev72, 7.10.1-rev32, 7.10.2-rev29, 7.10.3-rev15
Vendor notification: 2020-06-03
Solution date: 2020-07-01
Public disclosure: 2020-10-13
CVE reference: CVE-2020-15002
CVSS: 5.0 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:N/A:N)

Vulnerability Details:
Messaging account URLs can be set and requested through API. These are not filtered through our blacklists and may
contain local or internal network hosts.

Risk:
While this feature is not exposed through our user interface, knowledgable attackers can use the API to query internal
resources through network requests and assess availability of systems and what services they run. This can be used as
a
reconnaissance step during an attack.
```

Steps to reproduce:
1. Use the "/ajax/messaging/account" API to set up a new messaging account and provide an internal "url"
2. Use the "/ajax/messaging/message" message API to list new messages for this account. Based on the response time and error message it's possible to assess if a service is available or not.

Solution:
We extended existing blacklist checks to this feature.

Internal reference: OXUIB-308
Vulnerability type: Cross-site scripting (CWE-80)
Vulnerable version: 7.10.2 and 7.10.3
Vulnerable component: frontend
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.2-rev26, 7.10.3-rev13
Vendor notification: 2020-06-10
Solution date: 2020-07-01
Public disclosure: 2020-10-13
Researcher Credits: Zeeshan Khalid
CVE reference: CVE-2020-15004
CVSS: 4.3 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N)

Vulnerability Details:
Bootstrap attributes can be used to execute script code at appointment titles.

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 either send a
malicious calendar invite or be part of the same organization to invite the victim.

Steps to reproduce:
1. Create vacation with HTML code and Bootstrap attributes, which contain script code
2. Invite the victim and make her open the appointmnet pop-up view

Solution:
Sanitization has been improved to remove those attributes.

Internal reference: DOCS-2147
Vulnerability type: Server-side request forgery (CWE-918)
Vulnerable version: 7.10.2 and 7.10.3
Vulnerable component: documentconverter
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.2-rev6, 7.10.3-rev7
Vendor notification: 2020-05-07
Solution date: 2020-07-01
Public disclosure: 2020-10-13
Researcher Credits: Sreejith Krishnan R(@skr0x1c0)
CVE reference: CVE-2020-15002
CVSS: 5.0 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:N/A:N)

Vulnerability Details:
When adding images to documents through /appsuite/api/oxodocumentfilter&action=addfile, a number of checks are executed
and redirects are followed. As this takes some time the API response delay can be used to measure if a port is open or not.

Risk:
Attackers can use the API to query internal resources through network requests and assess availability of systems and what services they run. This can be used as a reconnaissance step during an attack.

Steps to reproduce:
1. Use the API to provide various URLs as external images
2. Check the time required for the API to reject the image

Solution:
We now use existing blacklist and URL resolution techniques to make sure redirects are not followed, which makes timing
attacks less reliable.

Internal reference: DOCS-2148
Vulnerability type: Server-side request forgery (CWE-918)
Vulnerable version: 7.10.2 and 7.10.3
Vulnerable component: documentconverter
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.10.2-rev6, 7.10.3-rev7
Vendor notification: 2020-05-07
Solution date: 2020-07-01
Public disclosure: 2020-10-13
Researcher Credits: Sreejith Krishnan R(@skr0x1c0)
CVE reference: CVE-2020-15002
CVSS: 4.9 (CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:C/C:L/I:N/A:L)

Vulnerability Details:
When adding images to documents through /appsuite/api/oxodocumentfilter&action=addfile, a DNS lookup is performed on the provided URL for external images. The lenght of the URL is however not limited, which allows attackers to provide huge URLs that lead to a "time of check / time of use" issue and excessive use of system resources.

Risk:
By injecting multiple requests and timing those properly, attackers can bypass existing checksa dn assess availability of systems at a restricted network and what services they run. This can be used as a reconnaissance step during an attack.

Steps to reproduce:
1. Use a huge URL as external image for document converter
2. While the validation logic is busy dissecting the URL, trigger another request

Solution:
We have severely restricted the acceptable lenght of URLs to the suggestion made at RFC3986. This makes timing attacks less reliable.

Internal reference: DOCS-2368
Vulnerability type: Cross-site scripting (CWE-80)
Vulnerable version: 7.10.3 and earlier
Vulnerable component: office
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.8.4-rev11, 7.10.1-rev7, 7.10.2-rev6, 7.10.3-rev7
Vendor notification: 2020-06-10
Solution date: 2020-07-01
Public disclosure: 2020-10-13

```
Researcher Credits: Sreejith Krishnan R (@skr0x1c0)
CVE reference: CVE-2020-15004
CVSS: 5.4 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N)

Vulnerability Details:
XML properties of ODT and ODP sources are used to create frontend structures that represent comments. In some cases
those properties are not properly escaped, which allows injection of script code through malicious documents.

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 (sending mail, deleting data etc.). To exploit this an additional step is necessary
which
could be achieved through social engineering.

Steps to reproduce:
1. Create a malicious ODT or ODP file with script code as annotation
2. Make a user open this document in "Edit" mode within the browser

Solution:
We improved our sanitization and escaping techniques for this kind of data sources.


---


Internal reference: DOCS-2437
Vulnerability type: Cross-site scripting (CWE-80)
Vulnerable version: 7.10.3 and earlier
Vulnerable component: office
Report confidence: Confirmed
Solution status: Fixed by Vendor
Fixed version: 7.8.4-rev11, 7.10.1-rev7, 7.10.2-rev6, 7.10.3-rev7
Vendor notification: 2020-06-10
Solution date: 2020-07-01
Public disclosure: 2020-10-13
Researcher Credits: notoriousrip
CVE reference: CVE-2020-15004
CVSS: 5.4 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N)

Vulnerability Details:
Header and footer identifiers within OOXML content can be used to inject script code when editing a document.

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 need to make
the victim edit a malicious document.

Steps to reproduce:
1. Modify the XML structure of a OOXML document, look for r:id attributes
2. Add script code to the value of such attributes
3. Open the document in edit mode

Solution:
Sanitization has been improved to properly handle those attributes.
```

Attachment: `signature.asc`
Description: Message signed with OpenPGP