Marek Bejda
marek.bejda@us.army.mil
May 16 2012
Updated OCT 10th 2012

J A V A S C R I P T   I N J E C T I O N

Applies to:

Joint Knowledge Online (JKO)
Defense Acquisition University (DAU)
Other course systems that work on the SCORM system

1. **Description**

Javascript execution happens within the user's web browser, this by default makes client-server communication very unreliable and insecure. Any experienced user can open a developers toolkit, right click and inspect the page. Developer toolkits often have a built-in object inspector and a console. The console controls the javascript actions of the website through which users can edit html objects, execute JS commands, and analyze variables. Or they can just execute functions directly, which is the way this bugs works.

2. **Idea**

From my observations SCORM by default uses Flash, Javascript (js), Prototype, and a Java back-end. Because of complexity Flash is not capable of securely communicating to the Java back-end and this void is patched using Javascript AJAX calls. Flash executes a JS completion command which sends an Ajax command to the server, which updates the users progress as they make through the training. Because of the client-side handling of Javascript a user can simulate the Flash command and update the completion status themselves.

3. **Tools**

   a. A browser (Safari, Firefox, or Internet Explorer)

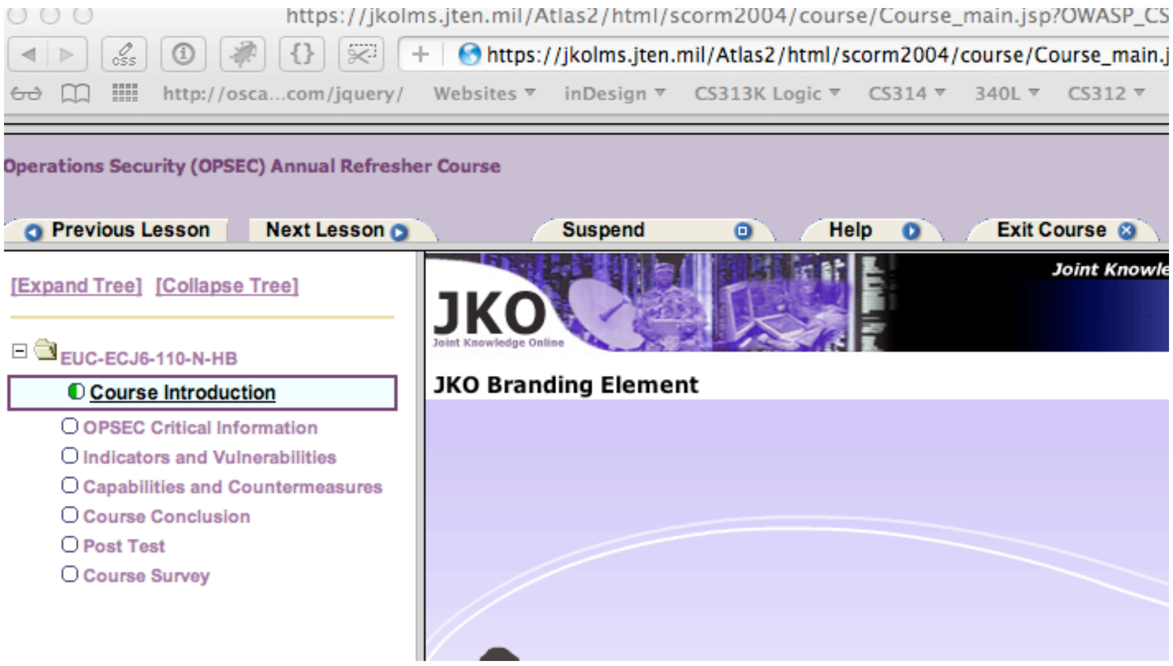
   b. The browsers developer's toolkit (Firebug, Internet Explorer Developer Tools, etc)

   c. Because Flash/JS/Java communication is not unique or encrypted, it can be duplicated by any user.

   d. For further documentation on the working of the system just read the .js files because they contain the development teams commentary.

4. **On to the exploit**

   a. Visit the SERE training on JKO or any other SCORM training page.


https://jkolms.jten.mil/Atlas2/html/scorm2004/course/Course_main.jsp?OWASP_CS
https://jkolms.jten.mil/Atlas2/html/scorm2004/course/Course_main.j
http://osca...com/jquery/
Websites
inDesign
CS313K Logic
CS314
340L
CS312
Previous Lesson
Next Lesson
Suspend
Help
Exit Course
[Expand Tree] [Collapse Tree]
JKO
Joint Knowledge Online
Joint Knowle

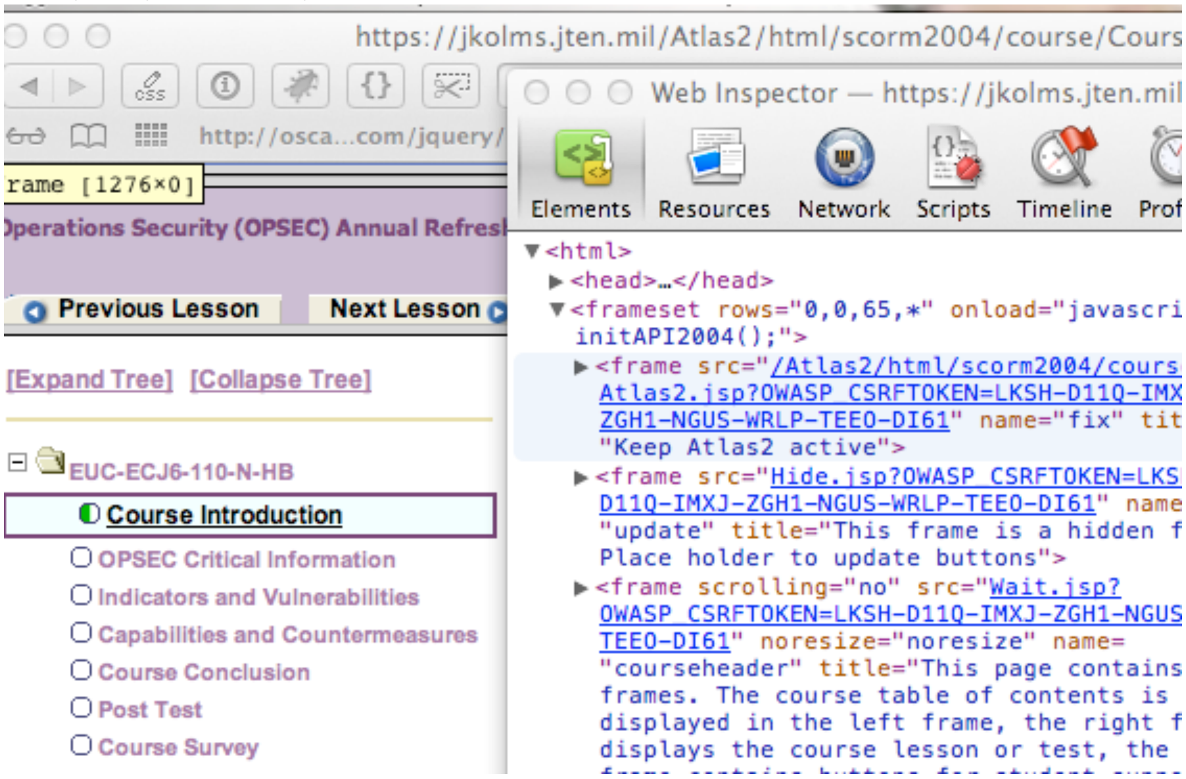
Operations Security (OPSEC) Annual Refresher Course

EUC-ECJ6-110-N-HB
- Course Introduction
- OPSEC Critical Information
- Indicators and Vulnerabilities
- Capabilities and Countermeasures
- Course Conclusion
- Post Test
- Course Survey

JKO Branding Element

b. Start the training.

c. Open up the development console.


https://jkolms.jten.mil/Atlas2/html/scorm2004/course/Cours
Web Inspector — https://jkolms.jten.mil
http://osca...com/jquery/
rame [1276×0]
Elements
Resources
Network
Scripts
Timeline
Prof
Previous Lesson
Next Lesson
[Expand Tree] [Collapse Tree]

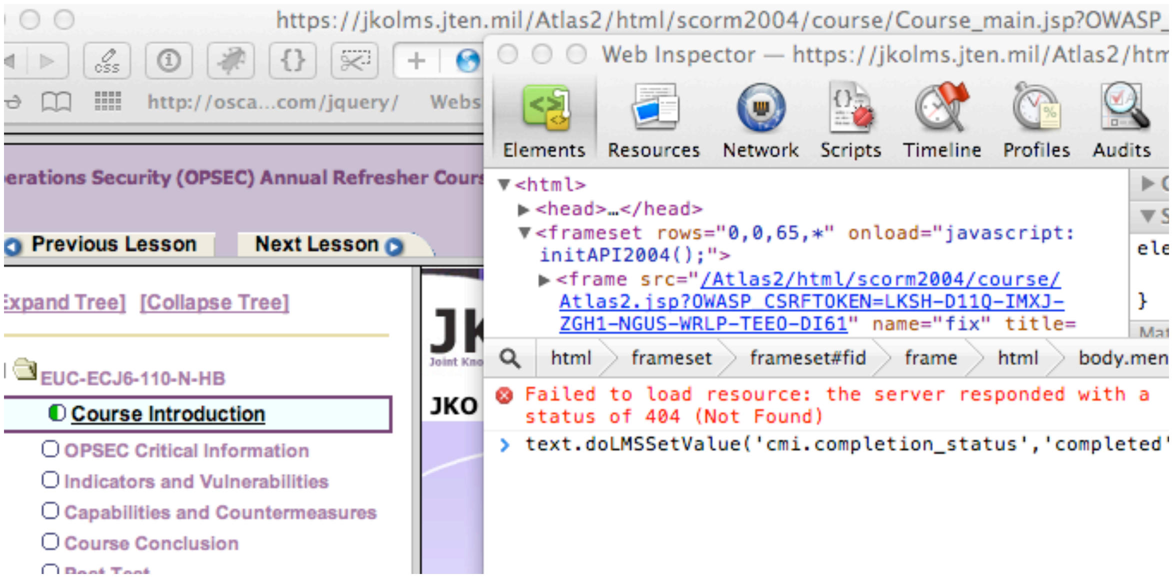
Operations Security (OPSEC) Annual Refres

EUC-ECJ6-110-N-HB
- Course Introduction
- OPSEC Critical Information
- Indicators and Vulnerabilities
- Capabilities and Countermeasures
- Course Conclusion
- Post Test
- Course Survey

```
▼<html>
 ▶<head>…</head>
 ▼<frameset rows="0,0,65,*" onload="javascri
   initAPI2004();">
   ▶<frame src="/Atlas2/html/scorm2004/cours
     Atlas2.jsp?OWASP_CSRFTOKEN=LKSH-D11Q-IMX
     ZGH1-NGUS-WRLP-TEEO-DI61" name="fix" tit
     "Keep Atlas2 active">
   ▶<frame src="Hide.jsp?OWASP_CSRFTOKEN=LKS
     D11Q-IMXJ-ZGH1-NGUS-WRLP-TEEO-DI61" name
     "update" title="This frame is a hidden f
     Place holder to update buttons">
   ▶<frame scrolling="no" src="Wait.jsp?
     OWASP_CSRFTOKEN=LKSH-D11Q-IMXJ-ZGH1-NGUS
     TEEO-DI61" noresize="noresize" name=
     "courseheader" title="This page contains
     frames. The course table of contents is
     displayed in the left frame, the right f
     displays the course lesson or test, the
```

d. In the console type in

text.doLMSSetValue('cmi.completion_status','completed');

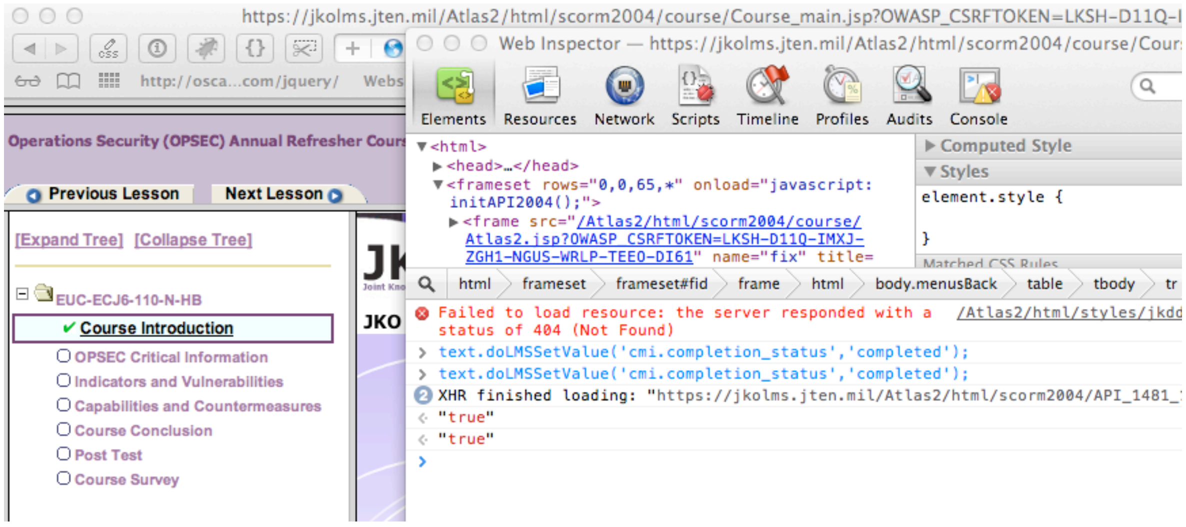

https://jkolms.jten.mil/Atlas2/html/scorm2004/course/Course_main.jsp?OWASP_
Web Inspector — https://jkolms.jten.mil/Atlas2/htm
http://osca...com/jquery/
Elements
Resources
Network
Scripts
Timeline
Profiles
Audits
erations Security (OPSEC) Annual Refresher Cour
Previous Lesson
Next Lesson
xpand Tree] [Collapse Tree]
EUC-ECJ6-110-N-HB
Course Introduction
OPSEC Critical Information
Indicators and Vulnerabilities
Capabilities and Countermeasures
Course Conclusion
JKO
html
frameset
frameset#fid
frame
html
body.men

```
▼<html>
  ▶<head>…</head>
  ▼<frameset rows="0,0,65,*" onload="javascript:
    initAPI2004();">
    ▶<frame src="/Atlas2/html/scorm2004/course/
      Atlas2.jsp?OWASP_CSRFTOKEN=LKSH-D11Q-IMXJ-
      ZGH1-NGUS-WRLP-TEEO-DI61" name="fix" title=
```

```
⊗ Failed to load resource: the server responded with a
  status of 404 (Not Found)
> text.doLMSSetValue('cmi.completion_status','completed'
```

e. The command sends an Ajax request to the server notifying of a lesson completion.

f. Wait a couple of seconds for the Ajax reply and the status symbol on the left should turn into a check.


https://jkolms.jten.mil/Atlas2/html/scorm2004/course/Course_main.jsp?OWASP_CSRFTOKEN=LKSH-D11Q-I
Web Inspector — https://jkolms.jten.mil/Atlas2/html/scorm2004/course/Cour
http://osca...com/jquery/
Elements
Resources
Network
Scripts
Timeline
Profiles
Audits
Console
Operations Security (OPSEC) Annual Refresher Cour
Previous Lesson
Next Lesson
[Expand Tree] [Collapse Tree]
EUC-ECJ6-110-N-HB
✓ Course Introduction
OPSEC Critical Information
Indicators and Vulnerabilities
Capabilities and Countermeasures
Course Conclusion
Post Test
Course Survey
JKO
Computed Style
Styles
Matched CSS Rules
html
frameset
frameset#fid
frame
html
body.menusBack
table
tbody
tr

```
▼<html>
  ▶<head>…</head>
  ▼<frameset rows="0,0,65,*" onload="javascript:
    initAPI2004();">
    ▶<frame src="/Atlas2/html/scorm2004/course/
      Atlas2.jsp?OWASP_CSRFTOKEN=LKSH-D11Q-IMXJ-
      ZGH1-NGUS-WRLP-TEEO-DI61" name="fix" title=
```

```
element.style {

}
```

```
⊗ Failed to load resource: the server responded with a   /Atlas2/html/styles/jkdd
  status of 404 (Not Found)
> text.doLMSSetValue('cmi.completion_status','completed');
> text.doLMSSetValue('cmi.completion_status','completed');
2 XHR finished loading: "https://jkolms.jten.mil/Atlas2/html/scorm2004/API_1481_
< "true"
< "true"
>
```

g. Lesson complete, repeat until the entire course is finished.

5. **Further notes**

Testing further I found additional loopholes, throughout DAU. If the above method does not work the course probably uses a different version of the SCORM API and the same can be accomplished using these commands:

-API.LMSSetValue('cmi.core.lesson_status','completed');

-text.setValue(‘cmi.lesson_status’,‘completed’);

API_1484_11.SetValue(‘cmi.completion_status’,‘completed’);

-API.LMSSetValue does not visibly change the status and the user has to change pages to see the result, unlike the first option (text.setLMSSetValue)

6. **Conclusion**

There are numerous ways to prevent the client from using a javascript injection to manipulate back-end data. Security+ teaches to never trust client-side input,clearly SCORM does not check authenticity or integrity of the data it receives. Another powerful programming practice is to minimize JS files to prevent unauthorized personnel from being able to read the plain JS files. The course completion injection is not even the top of the iceberg as SCORM receives and transmits users data, but I have not had time yet to experiment with altering other peoples profiles.