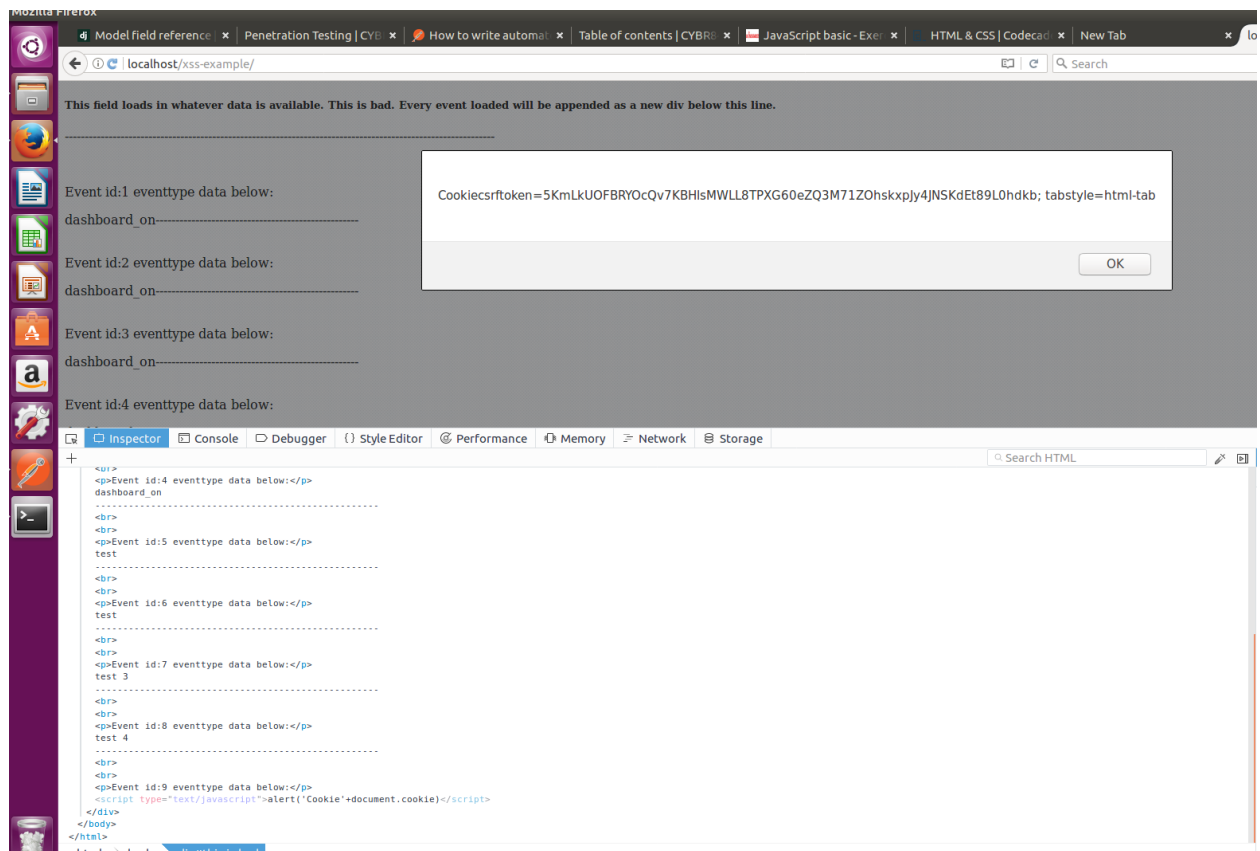


The left screenshot shows a web application interface with a blue event ticket icon. The ID field contains the payload: `(ID: 13) <script type='text/javascript'>alert('Cookie'+document.cookie)</script>`. The right screenshot shows the same interface with the payload executed, displaying the cookies in an alert box: `Cookiecsrftoken=5KmLkUOFBRYOcQv7KBHIsMWLL8TPXG60eZQ3M71ZOhsxpy4jN5KdEt89L0hdkb; tabstyle=html-tab`.

Below the screenshots, a browser window is shown with the URL `localhost/xss-example/`. The page content includes a warning: "This field loads in whatever data is available. This is bad. Every event loaded will be appended as a new div below this line." followed by a list of event IDs and their corresponding data. An alert box is displayed, showing the cookies: `Cookiecsrftoken=5KmLkUOFBRYOcQv7KBHIsMWLL8TPXG60eZQ3M71ZOhsxpy4jN5KdEt89L0hdkb; tabstyle=html-tab`.



The screenshot displays the Postman application interface. On the left, the 'Collections' tab is active, showing a collection named 'Demo REST API Unit Tests' with 2 requests. The first request, 'Persistent XSS test', is selected. The main panel shows the details of this request, which is a POST to 'http://localhost/api/activateifttt'. The 'Tests' tab is selected, showing a JavaScript test script:

```
1 var jsonData = JSON.parse(responseBody);
2 tests["XSS Prevented"] = jsonData.success !== false;
```

Below the script, the test results are shown as 'FAIL XSS Prevented'. The interface also includes a top bar with 'Runner', 'Import', and 'Builder' tabs, and a bottom bar with 'Body', 'Cookies', 'Headers (7)', and 'Tests (0/1)' tabs.

The screenshot shows the Postman interface with a collection named "Demo REST API Unit Tests" containing 2 requests. The selected request is a POST to "http://localhost/api/session". The "Tests" tab is active, showing a JavaScript test script:

```
1 var jsonData = JSON.parse(responseBody);
2 postman.setGlobalVariable("isauthenticated", jsonData.isauthenticated);
```

Below the script, the "Cookies" tab is active, displaying a table of cookies:

Name	Value	Domain
sessionId	3imk8zlteqegqmc vjv37v9tkhauwpb 0q	localhost

The screenshot shows the Postman interface with the same collection. The selected request is a POST to "http://localhost/api/events". The "Tests" tab is active, showing a JavaScript test script:

```
1 var jsonData = JSON.parse(responseBody);
2 tests["Event Endpoint Authentication Check"] = (jsonData.success === true) && JSON.parse(globals.isauthenticated);
3
4
```

Below the script, the "Body" tab is active, displaying the response in JSON format:

```
1 {
2   "success": true
3 }
```

The screenshot displays the Postman application interface. On the left sidebar, the 'Collections' tab is active, showing a collection named 'Demo REST API Unit Tests' containing two requests: 'Persistent XSS test' and 'Auth Check'. The 'Persistent XSS test' request is selected and expanded in the main panel.

The request details for 'Persistent XSS test' are as follows:

- Method:** POST
- URL:** http://localhost/api/activateifttt
- Body Type:** raw (selected)
- Body Content:**

```
1 {
2   "eventtype": "<script type='text/javascript'>alert('Cookie'+document.cookie)</script>",
3   "timestamp": 1500683745,
4   "userid": "test6"
5 }
```

Below the request editor, the 'Body' tab is selected, showing the response body in JSON format:

```
1 {
2   "success": true
3 }
```

The image is a composite of three screenshots related to a penetration test.

Top Screenshot: A code editor showing a Django model definition in `models.py`. The code defines an `Event` model with fields for `eventtype`, `timestamp`, `userid`, and `requestor`. It also includes a `__str__` method and an `EventAdmin` class.

```
1 from __future__ import unicode_literals
2
3 from django.db import models
4 from django.core.validators import *
5
6 from django.contrib.auth.models import User, Group
7
8 from django.contrib import admin
9 import base64
10 from django_bleach.models import BleachField
11
12 class Event(models.Model):
13     #eventtype = models.CharField(max_length=1000, blank=False)
14     eventtype = BleachField()
15     timestamp = models.DateTimeField()
16     userid = models.CharField(max_length=1000, blank=True)
17     requestor = models.GenericIPAddressField(blank=False)
18
19     def __str__(self):
20         return str(self.eventtype)
21
22 class EventAdmin(admin.ModelAdmin):
```

Bottom Left Screenshot: A REST client showing a JSON response for a GET request to `http://localhost/api/events/`. The response is a list of event objects.

```
{
  "timestamp": "1970-01-18T08:51:23Z",
  "userid": "admin",
  "requestor": "172.19.0.1"
},
{
  "model": "api.event",
  "pk": 10,
  "fields": {
    "eventtype": "<script type='text/javascript'>alert('XSS Attack!');</script>",
    "timestamp": "1970-01-18T08:51:23Z",
    "userid": "admin",
    "requestor": "172.19.0.1"
  }
},
{
  "model": "api.event",
  "pk": 11,
  "fields": {
    "eventtype": "<script type='text/javascript'>alert('XSS Attack!');</script>",
    "timestamp": "1970-01-18T08:51:23Z",
    "userid": "admin",
    "requestor": "172.19.0.1"
  }
},
{
  "model": "api.event",
  "pk": 12,
  "fields": {
    "eventtype": "Pen test lab",
    "timestamp": "1970-01-18T08:51:21Z",
    "userid": "pentester",
    "requestor": "172.19.0.1"
  }
},
{
  "model": "api.event",
  "pk": 13,
  "fields": {
    "eventtype": "<script type='text/javascript'>alert('XSS Attack!');</script>",
    "timestamp": "1970-01-18T08:51:23Z",
    "userid": "admin",
    "requestor": "172.19.0.1"
  }
},
{
  "model": "api.event",
  "pk": 14,
  "fields": {
    "eventtype": "unit-test-events",
    "timestamp": "1970-01-18T08:51:21Z",
    "userid": "rex",
    "requestor": "172.19.0.1"
  }
},
{
  "model": "api.event",
  "pk": 15,
  "fields": {
    "eventtype": "<script type='text/javascript'>alert('Cookie'+document.cookie)</script>",
    "timestamp": "1970-01-18T08:51:23Z",
    "userid": "admin",
    "requestor": "172.19.0.1"
  }
}
]
```

Bottom Right Screenshot: A web application interface titled "cybr 8470 demo app". It shows a "Dashboard Demo App" with a "Logged in as: rex (Logout)" message. A security alert is displayed: `(ID: 15) <script type='text/javascript'>alert('Cookie'+document.cookie)</script>`. The alert is dated "Sunday January 18th 1970 2:51:23 am".

Event id:15 eventtype data below:

```
<script type="text/javascript">alert('Cookie'+document.cookie)</script>-----
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
    "pk": 14,
    "fields": {
      "eventtype": "unit-test-events",
      "timestamp": "1970-01-18T08:51:21Z",
      "userid": "rex",
      "requestor": "172.19.0.1"
    }
  },
  "model": "api.event",
  "pk": 15,
  "fields": {
    "eventtype": "&lt;script type='text/javascript'&gt;alert('Cookie'+document.cookie)&lt;/script&gt;",
    "timestamp": "1970-01-18T08:51:23Z",
    "userid": "admin",
    "requestor": "172.19.0.1"
  }
}
```

The screenshot shows the 'Collection Runner' application with the 'Run Results' tab selected. The test suite is 'Demo REST API Unit Tests' with 0 passed and 2 failed tests. The results are as follows:

Iteration	Test Name	URL	Path	Status	Response	Time	Size
Iteration 1	POST Persistent XSS test	http://localhost/api/activ...	...API Unit Tests / Persistent XSS test	200 OK	429 ms	16 B	
	FAIL XSS Prevented						
	GET Auth Check	http://localhost/api/sessi...	...o REST API Unit Tests / Auth Check	200 OK	28 ms	55 B	
	This request does not have any tests.						
	POST Event Endpoint Authentication Check	http://localhost/api/events	..ent Endpoint Authentication Check	200 OK	39 ms	16 B	
	FAIL Event Endpoint Authentication Check						

Code changes for Django bleach can be found at - <https://github.com/akhampariya/CYBR8470-building-a-webservice-lab/tree/my-work>