What Is Request Forgery

- No need to figure user’s session ID
- Uses the user’s existing browser session
- Tricks the user into submitting a request
  - Create
  - Delete
  - Update or Change

Both OSRF and CSRF rely on the following:

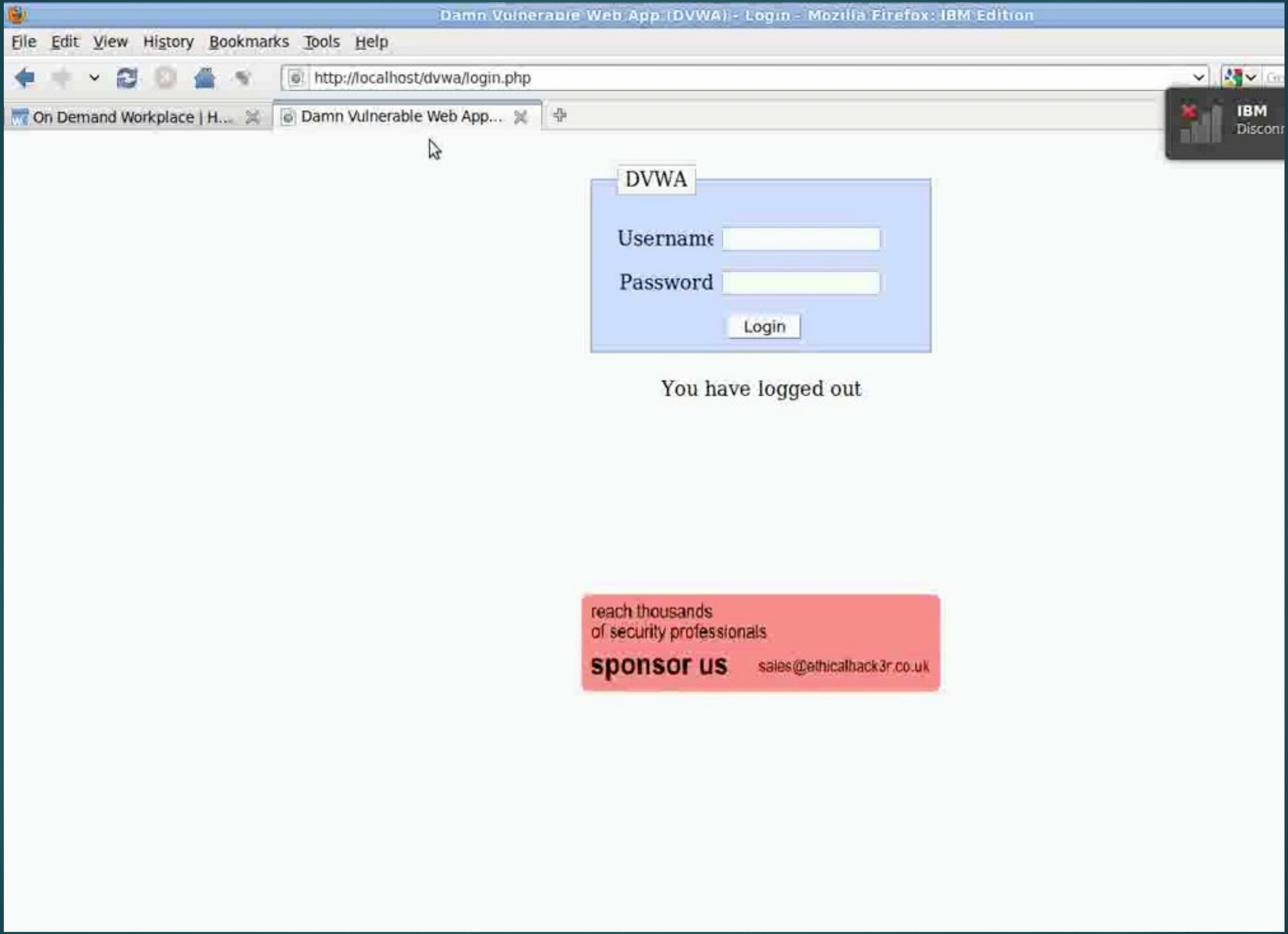
1. Web browser handing of session related information:
   - Cookies and HTTP authentication info.
2. Knowledge of valid web application URLs
3. Web page functions which rely on Cookies and HTTP authentication info known by the web browser
4. The existence of HTML elements which call back to server resources.
   - e.g.: an image tag
     <img src="http://example.com"/>

What is OSRF?

- On
- Site
- Request
- Forgery

What is CSRF?

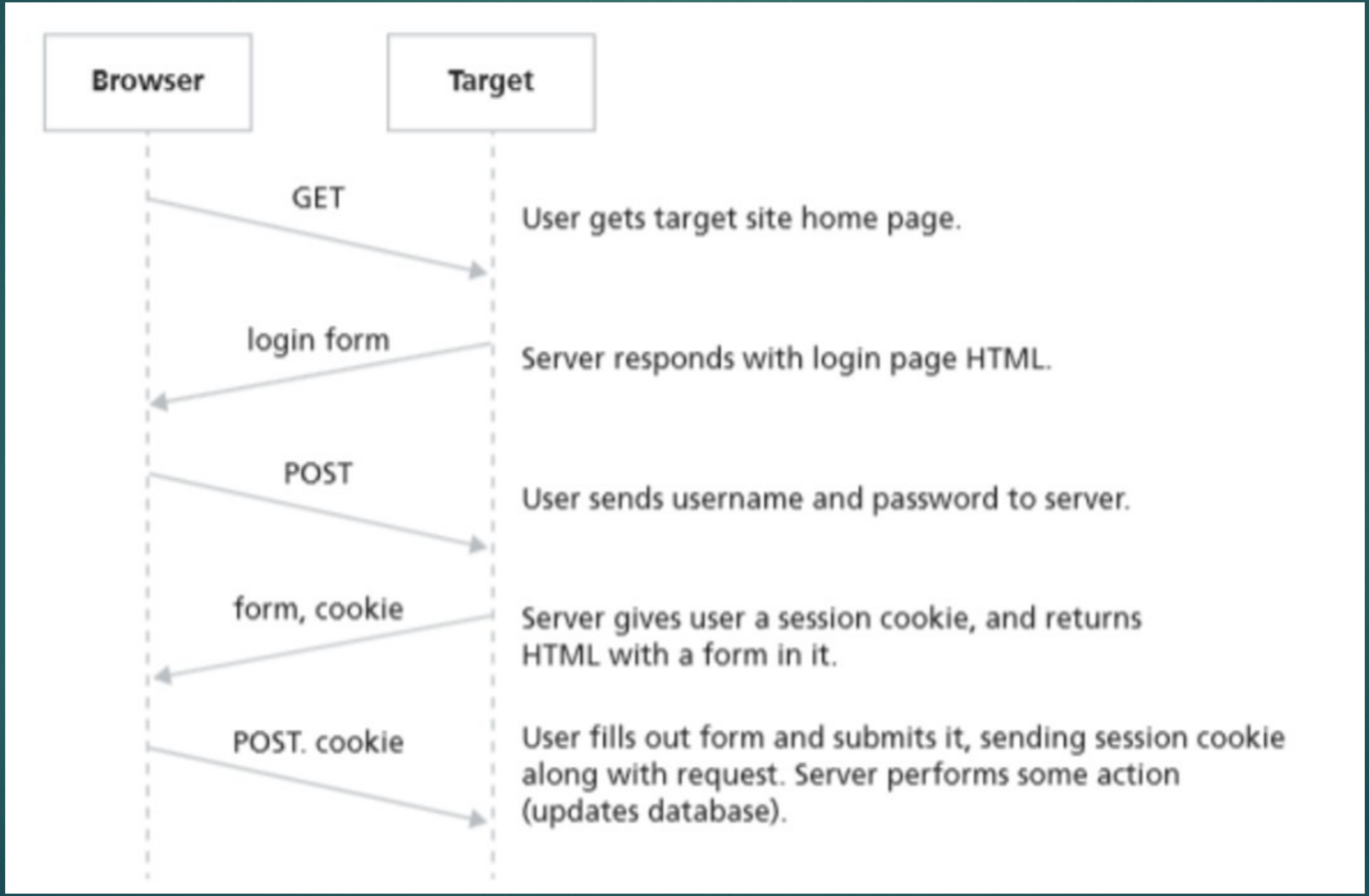
- Cross
- Site
- Request
- Forgery


Damn Vulnerable Web App (DVWA) - Login - Mozilla Firefox: IBM Edition
File Edit View History Bookmarks Tools Help
http://localhost/dvwa/login.php
On Demand Workplace | H...
Damn Vulnerable Web App...
IBM
Disconn
DVWA
Username
Password
Login

You have logged out

reach thousands
of security professionals
sponsor us sales@ethicalhack3r.co.uk

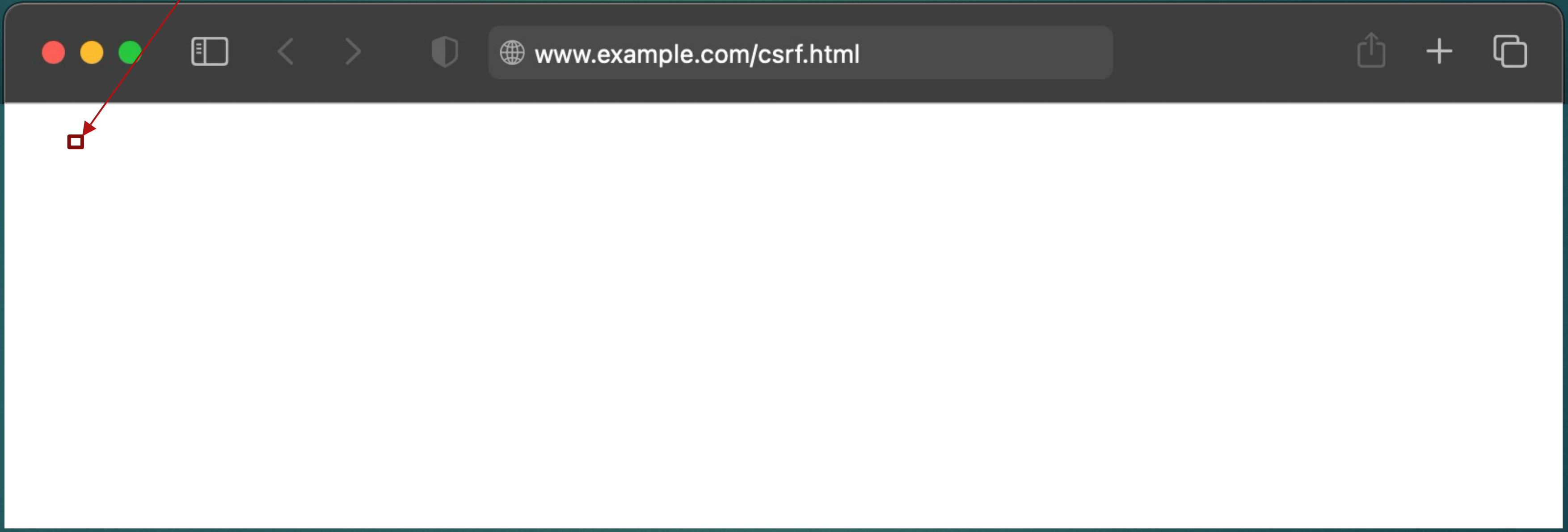
CSRF
User – Browser – Server Interaction



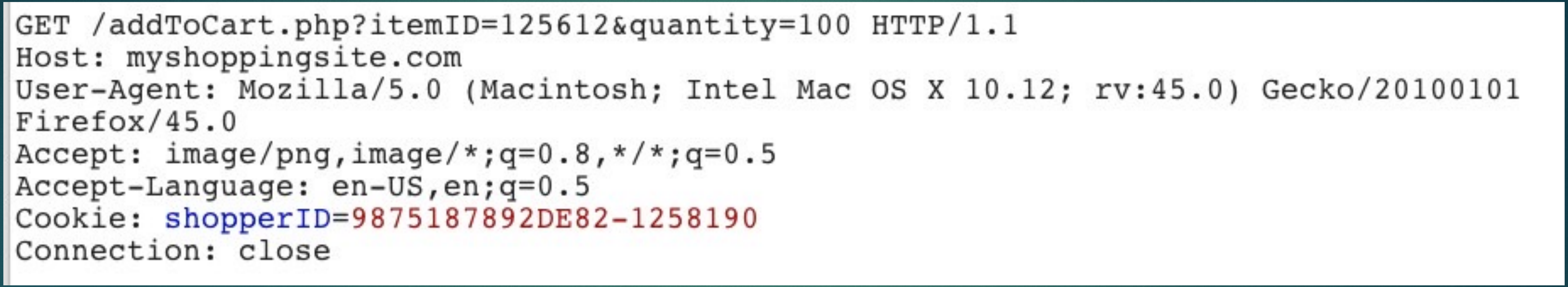
Browser
Target
GET
User gets target site home page.
login form
Server responds with login page HTML.
POST
User sends username and password to server.
form, cookie
Server gives user a session cookie, and returns HTML with a form in it.
POST. cookie
User fills out form and submits it, sending session cookie along with request. Server performs some action (updates database).

Example #1
GET CSRF HTML

```
<img width="1" height="1" src="https://myShoppingSite.com/addToCart.php?itemID=125612&quantity=100">
```

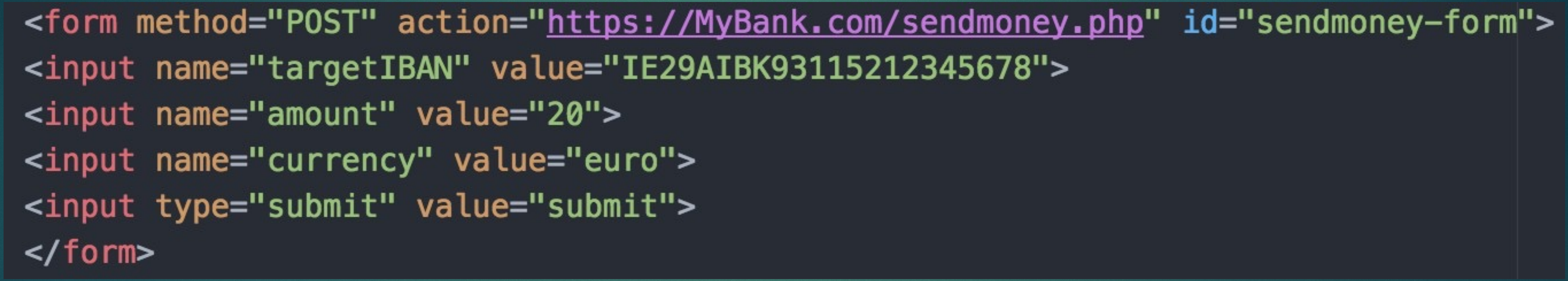


GET CSRF HTTP Request



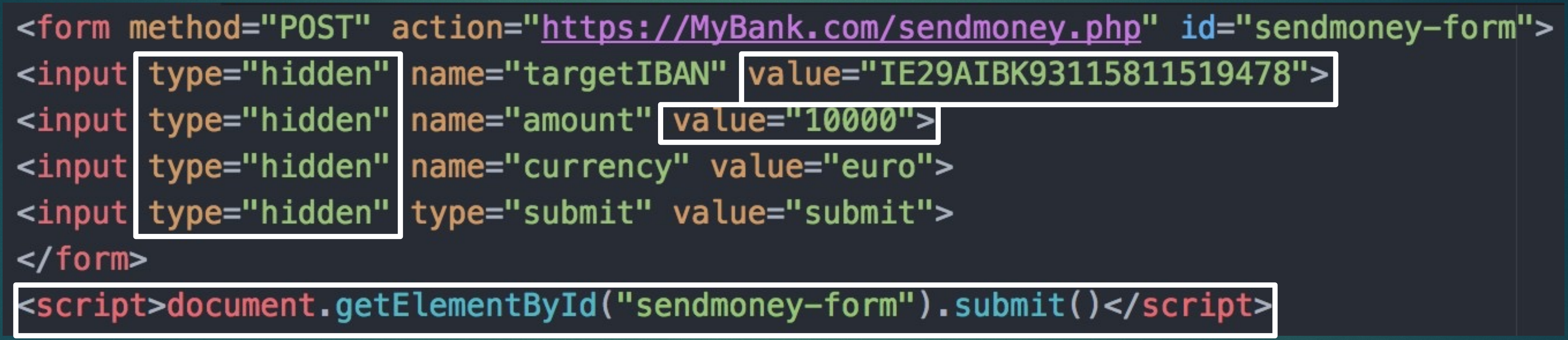
```
GET /addToCart.php?itemID=125612&quantity=100 HTTP/1.1
Host: myshoppingsite.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:45.0) Gecko/20100101
Firefox/45.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Cookie: shopperID=9875187892DE82-1258190
Connection: close
```

Example #2
CSRF vulnerable POST HTML Form



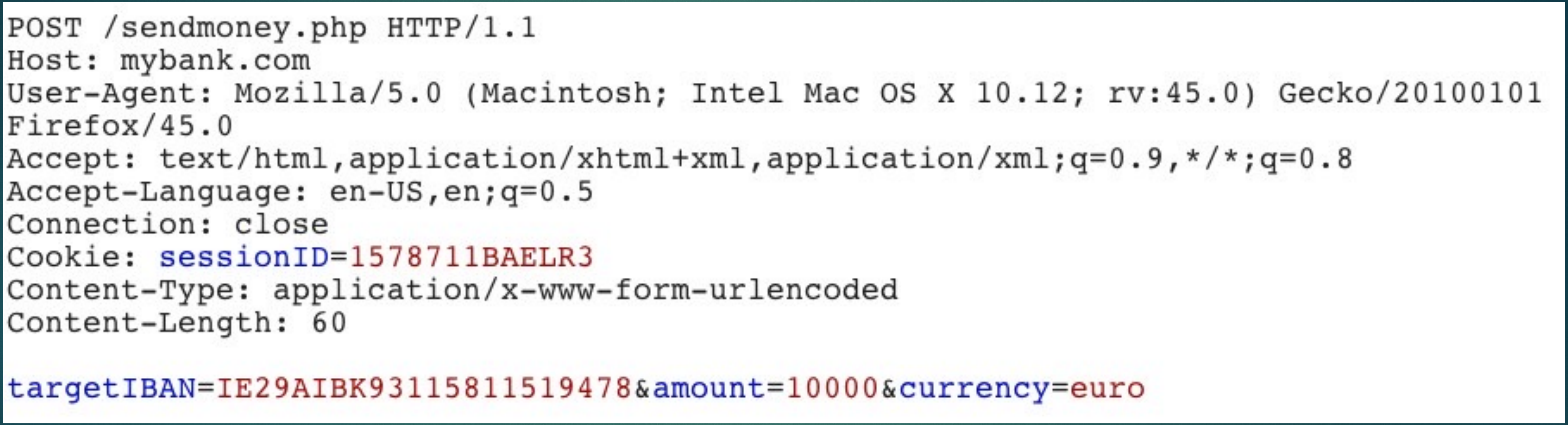
```html
<form method="POST" action="https://MyBank.com/sendmoney.php" id="sendmoney-form">
<input name="targetIBAN" value="IE29AIBK93115212345678">
<input name="amount" value="20">
<input name="currency" value="euro">
<input type="submit" value="submit">
</form>
```

Example #2
CSRF vulnerable POST HTML Form



```
<form method="POST" action="https://MyBank.com/sendmoney.php" id="sendmoney-form">
<input type="hidden" name="targetIBAN" value="IE29AIBK93115811519478">
<input type="hidden" name="amount" value="10000">
<input type="hidden" name="currency" value="euro">
<input type="hidden" type="submit" value="submit">
</form>
<script>document.getElementById("sendmoney-form").submit()</script>
```

Example #2
CSRF POST HTTP Request



```
POST /sendmoney.php HTTP/1.1
Host: mybank.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:45.0) Gecko/20100101
Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
Cookie: sessionID=1578711BAELR3
Content-Type: application/x-www-form-urlencoded
Content-Length: 60

targetIBAN=IE29AIBK93115811519478&amount=10000&currency=euro
```

How to test for CSRF

Black Box Testing

1. Know the URLs
   - Two users useful for testing.
2. Pick a URL and then create a HTML page containing the http request referencing the URL
3. Ensure the Victim user is logged in
4. Find a method to have the Victim click or follow the link (some social engineering may be involved)
5. Observe the result

How to test for CSRF

Grey Box testing

1. Review the web application to determine if its session management is vulnerable.
2. If session management only relies on client side values (values only determined on the loaded web page)
    - Cookies
    - HTTP authentication credentials

CSRF Mitigation Do’s

1. Generate a unique **nonce** for each form.
   - Per-session nonce
   - Per-request nonce
2. Same origin policy HTTP header.

CSRF Mitigation Do Not’s

- Using a Secret Cookie.
- Only Accepting POST Requests
- Multi-Step Transactions
- URL Rewriting
- HTTPS

How to protect yourself from CSRF attacks

- Remember to log out of applications
- Change default passwords
- Use different browsers for sensitive browsing
- Use a Virtual Machine