Web (2.0) Security

Ben Adida
CRCS Seminar
18 April 2007
web browsers in 1999 were **dumb clients**.

web security was about protecting the **server** from a **rogue client**.
web browsers in **2007** are *platforms.*

web security is about protecting *applications* from *each other.*
It is worth understanding this platform, our de-facto Trusted Computing Base.

Installed Base

> 1 Billion
Evolution of Web Security

- Basics of HTTP/HTML
- Web 1.0 Security Issues
- Advanced HTTP/HTML/Javascript
- Web 2.0 Security Issues
- A Few Proposals
SSL?

“Using encryption on the Internet is the equivalent of arranging an armored car to deliver credit card information from someone living in a cardboard box to someone living on a park bench.”

Gene Spafford
OS
Password Cracks

Web Server

Database

OS
Password Cracks

Web Server

Database

OS

Buffer Overflow, etc..
Password Cracks

Application Code

Web Server

Database

OS

Buffer Overflow, etc..
Password Cracks

Buffer Overflow, etc..

Web Server

Database

OS

Application Code
Application-Level Attacks

Application Code

Web Server

Database

OS

Password Cracks

Buffer Overflow, etc..
Basics of HTTP
GET
/2007/TECH/space/
04/17/
space.shuttle.ap/
index.html
CAPE CANAVERAL, Florida (AP) -- NASA will try...
WordPress Blog

GET
/?p=218

User

<div class="entry">
<p>Speaker: Ben Adida, Harvard</p>
</div>
POST
/send_email

to=ben@eecs.harvard.edu
&subject=Talk
&body=...
POST /send_email
  to=ben@eecs.harvard.edu
  &subject=Talk
  &body=...

<div>
  message sent.
<br />
</div>
GET /
/article
GET /article

Set-Cookie: uid=12345

Cookie: uid=12345

User
Web 1.0 Security
The web site is an API with a security policy.

Can the API be hijacked?
Discovering Secret API Calls

Harvard rejects 119 accused of hacking
Applicants' behavior 'unethical at best'
By Robert Weisman, Globe Staff | March 8, 2005

The Boston Globe
Discovering Secret API Calls

Harvard rejects 119 accused of hacking

Applicants' behavior 'unethical at best'
By Robert Weisman, Globe Staff | March 8, 2005

Business ethics 101: in the event of a blatant IT failure, blame the user.
Illegal Inputs
Illegal Inputs

http://acme.com/view-user?id=4
Illegal Inputs

http://acme.com/view-user?id=4

select name, email from users
where USERID = $id;
Illegal Inputs

http://acme.com/view-user?id=4

select name, email from users
where USERID = $id;

view-user?id=nullable union select
password as name, email from users
where USERID=4
Illegal Inputs

http://acme.com/view-user?id=4

select name, email from users
where USERID = $id;

view-user?id=NULL union select
password as name, email from users
where USERID=4

select name, email from users
where USERID = NULL union select password
as name, email from users where USERID=4;
Spoofed Inputs
Spoofed Inputs

<form action="buy-tickets">
<input
    type="hidden"
    name="adult_price"
    value="9.5" />
...
</form>
Spoofed Inputs

Happens when the server-side software stack provides (poor) high-level abstractions.

```html
<form action="buy-tickets">
  <input type="hidden" name="adult_price" value="9.5" />
  ...
</form>
```
Third-Party Cookies
Third-Party Cookies
Third-Party Cookies

Enhance your identity theft protection with Equifax.
Daily credit monitoring. Identity theft alerts. And more.

Find out more
Ads are now delivered in IFRAMES, which can get around third-party restrictions.
Advanced HTTP/HTML/Javascript
Web Pages are Programs
Web Pages are Programs

<head>
<title>Random Blog</title>

<script src="http://www.google-analytics.com/urchin.js" type="text/javascript">
</script>
</head>
Introspection

Network Requests

```javascript
var req = new XMLHttpRequest();

req.onreadystatechange = function() {
    if (req.readyState == 4) {
        // do something with req.responseText
    }
}

req.open("GET", "/more-data", true);
req.send(""");
```
Introspection

```javascript
document.getElementById('article_content').innerHTML;
```

Network Requests

```javascript
var req = new XMLHttpRequest();

req.onreadystatechange = function() {
  if (req.readyState == 4) {
    // do something with req.responseText
  }
};

req.open("GET", "/more-data", true);
req.send("");
```

Modifying the Page

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

var i = document.createElement('img');
i.src = 'http://otherdomain.com/photo2.jpg';
document.body.appendChild(i);
```
BigBank.com Cookie
sessionid=8d8c93200
User

BigBank.com

MyFriendBook.com

Cookie

sessionid=8d8c93200

GET
/friends
BigBank.com Cookie
sessionid=8d8c93200

GET /friends

Bank balance
stealing code

User
BigBank.com

GET /balance

MyFriendBook.com

GET /friends

Bank balance stealing code

BigBank.com Cookie
sessionId=8d8c93200

User
BigBank.com

MyFriendBook.com

GET /friends

GET /balance

$42

Bank balance stealing code

BigBank.com Cookie
sessionid=8d8c93200

User
BigBank.com

GET /balance

$42

User

MyFriendBook.com

GET /friends

Bank balance stealing code

$42

BigBank.com Cookie
sessionid=8d8c93200
BigBank.com

GET /friends

MyFriendBook.com

GET /friends

User

BigBank.com Cookie
sessionid=8d8c93200

Bank balance stealing code

GET /balance

$42
Intranet

MyFriendBook.com

User
Intranet

User

MyFriendBook.com

GET /friends

Port scan code
Intranet

User

MyFriendBook.com

GET /friends

Port scan code

GET /employees
Intranet

User

MyFriendBook.com

GET /friends

Port scan code

"Alice, Bob,..."

GET /employees

"Alice, Bob,..."
Intranet

GET /employees

"Alice, Bob,..."

GET /friends

Port scan code

"Alice, Bob,..."

User

MyFriendBook.com
Cross-Domain Images

flickr.com

MyBlog.com

GET /post?id=3
Blog Post w/ Inline image

User
Cross-Domain Images

flickr.com

MyBlog.com

GET /post?id=3
Blog Post w/ Inline image

GET /photo

User
Cross-Domain Images

---

**flickr.com**

GET /photo

**MyBlog.com**

GET /post?id=3

Blog Post w/ Inline image

User
Cross-Domain Scripts
<head>
  <title>Random Blog</title>

  <script src="http://www.google-analytics.com/urchin.js" type="text/javascript">
  </script>

</head>
Cross-Domain Scripts

```html
<head>
  <title>Random Blog</title>
  
  <script src="http://www.google-analytics.com/urchin.js" type="text/javascript">
  
  </script>

</head>

var new_script = document.createElement('script');
new_script.src = 'http://otherdomain.com/';
document.head.appendChild(new_script);
```
Cross-Domain Scripts

```html
<head>
  <title>Random Blog</title>
  <script src="http://www.google-analytics.com/urchin.js" type="text/javascript">
  </script>
</head>

var new_script = document.createElement('script');
new_script.src = 'http://otherdomain.com/';
document.head.appendChild(new_script);
```
State of Cross-Domain

• Cross-Domain outgoing messages can always be sent (using parameters in an IMG url or SCRIPT or submitted FORM)

• Cross-Domain incoming data can be integrated in limited ways:
  ➡ images rendered, but opaquely
  ➡ scripts executed, but opaquely
JSON and Callbacks
JSON and Callbacks

["ct","Alice","alice@gmail.com"],
["ct","Bob","bob@gmail.com"],
...
]
JSON and Callbacks

```javascript
[{
  "ct": "Alice", "alice@gmail.com",
},
{
  "ct": "Bob", "bob@gmail.com",
},
...
]

getJSON("https://example.com/data.json", function(data) {
  JSONP_CALLBACK(data);
});
```
JSON and Callbacks

```json
[
    ['ct', 'Alice', 'alice@gmail.com'],
    ['ct', 'Bob', 'bob@gmail.com'],
    ...
]
```

```javascript
JSONP_CALLBACK([  
    ['ct', 'Alice', 'alice@gmail.com'],
    ['ct', 'Bob', 'bob@gmail.com'],
    ...
]);
```

Easy data format which can be parsed using any programming language, especially JS. Hacked Cross-Domain!
Web 2.0 Security
Cross-Site Scripting (XSS)
Cross-Site Scripting (XSS)

Evil.org

MyFriendBook.com

Malicious Code posted on personal profile
Cross-Site Scripting (XSS)

Evil.org
MyFriendBook.com

GET /eve

Malicious Code posted on personal profile

User

Eve
Cross-Site Scripting (XSS)

Evil.org

GET /eve

MyFriendBook.com

Normal Page + Malicious Injected Code

Eve

Malicious Code posted on personal profile

User
Cross-Site Scripting (XSS)

POST or GET
/steal_cookie?
friendbook_id=3234

Malicious Code posted on personal profile

Normal Page + Malicious Injected Code

Evil.org
MyFriendBook.com

User
Eve
Cross-Site Request Forgery (CSRF)

BigBank.com  MyFriendBook.com

BigBank.com Cookie
sessionid=8d8c93200

User
Cross-Site Request Forgery (CSRF)

BigBank.com

MyFriendBook.com

GET
/eve

Malicious Code

BigBank.com Cookie
sessionid=8d8c93200

User
Cross-Site Request Forgery (CSRF)

BigBank.com

MyFriendBook.com

POST /transfer

GET /eve

Malicious Code

BigBank.com Cookie
sessionid=8d8c93200

User
CSRF + JavaScript/JSON

http://jeremiahgrossman.blogspot.com/
CSRF + JavaScript/JSON

[ ["ct","Alice","alice@gmail.com"],
  ["ct","Bob","bob@gmail.com"],
  ...
]
CSRF + JavaScript/JSON

```javascript
var new_script = document.createElement('script');
new_script.src = 'http://google.com/addressbook';
// override the Array constructor
Array = new function() {
    ...
};
document.head.appendChild(new_script);
```

http://jeremiahgrossman.blogspot.com/
History Insecurity

- Links already visited are colored differently in a web browser.
- Javascript can introspect...
- More targeted phishing.
Timing Information Leakage

- Cached images are quick to load.
- Referencing a BankOfAmerica image and checking how long it takes to load.
- Referencing SCRIPTs on an intranet address and seeing how long they take to “time out” to effectively port scan.
Meanwhile... Mashups!
Hand over your Password!

Check your address book to see who uses LinkedIn

Check webmail contacts

Don't use webmail?
Let me take over your web browser

“My del.icio.us” button
View and search your saved pages

Your saved pages are always available at:
http://del.icio.us/benadida
This is not a fluke. Mashups are the future.

How do we secure them?
Cross-Frame Messaging
Cross-Frame Messaging

http://acme.com/article#paragraph_2
Cross-Frame Messaging

http://acme.com/article#paragraph_2

http://acme.com/article#paragraph_3
Cross-Frame Messaging

http://acme.com/article#paragraph_2

http://acme.com/article#paragraph_3

Try the Windows Live Contacts control - tile view

This is a simple mashup that demonstrates the communication features in tile view. The publisher or developer can specify a text message to be sent when communication is started from this control.

Right click view source to see how all this is done - it's one page of javascript and html.

The presence icons to start instant messages only appear if:

- you are logged into Windows Live Messenger
- you are browsing in Internet Explorer

Even when the above is not met, you still get the email and contact card icons.

You don't need to give up a fixed area of your screen to the control, you can make it hide and show.

Hide and show the control
Idea #1: Expand “Platform” Features of the Browser.
Current Work

• JSONRequest
  http://www.json.org/JSONRequest.html

• Cross-Document Messaging

• MashupOS [JacksonWang2007]
  at HotOS2007.
Loosely Coupled Mashups

- **event announcement**
  extract and send to Calendar of my choice
  (Google Cal, Yahoo Cal, 30Boxes, ...)

- **login form**
  auto-fill with credentials provided
  by some identity provider.

- **“recommend” feature**
  prompt to enter a name from address book.
3 Simple Proposals for Loosely Coupled Mashups

- **Javascript Clean Slate**
  making sure the API is not hijacked.

- **Limited Extensions**
  extensions that awaken only when activated.

- **Metadata-Driven Extensions**
  extensions that process only metadata-driven requests.

[Adida2007]
Idea #2: Security in the App Layer
• Cache Cookies [JuelsJakobssonJagatic] using the cache to store secret tokens.

• Active Cookies [JuelsJakobssonStamm] using the cache to store Javascript verification code.

• Subspace [JacksonWang] using tricks of subdomains to isolate Javascript code.
Secure Tokens in the Fragment Identifier
Secure Tokens in the Fragment Identifier

http://myfriendbook.com/resource#8cvb8234234
Secure Tokens in the Fragment Identifier

http://myfriendbook.com/resource#8cvb8234234

Not sent over the network.
Secure Tokens in the Fragment Identifier

http://myfriendbook.com/resource#8cvb8234234

Not sent over the network.

Does not cause a page reload.
Closing Thoughts
Closing Thoughts

- The web browser is a platform.
Closing Thoughts

• The web browser is a platform.

• It needs cross-application features with a solid security model.
Closing Thoughts

• The web browser is a platform.

• It needs cross-application features with a solid security model.

• There are still interesting tricks to pull based on existing protocols and behaviors.
Questions?

http://flickr.com/photos/hollywoodpoodle/373053089/