SQL Injection

Can chain together statements, and can modify existing statements
Not Just SQL!

User

Forms a string containing user input

Front end

Executes this string as a command or query

Back end

Injection vulnerabilities are a generic issue!
PREVENTING INJECTION ATTACKS

Validate all the inputs!
Most injection attacks trick application into interpreting data as code.

This changes the semantics of a query or command generated by the application.

Make sure unsafe inputs cannot change the meaning of query.
A basic web architecture

Much of the user data is part of the browser

DB is a separate entity, logically (and often physically)
Where Does the Attacker Live?

Client

Browser: renders the webpage

Malware attacker

Private data

Server

Web Server hosts the web page

Network attacker

Web server attacker

Database

Much of the user data is part of the browser

DB is a separate entity, logically (and often physically)
Web Architecture: Simplified View

Client Side

- Document Renderer
- HTML Parser

Protocols

- HTML
- Gopher
- FTP
- HTTP

Server Side

- Network Protocols
- Web servers: Responsible for securely parsing input data
  - PHP, Ruby, ASP, JSP

Web Browser

- Responsible for securely confining Web content presented by visited websites
Overview

• The Web Model
  • What components make up today’s browsers and web servers?
  • How has this functionality evolved over time?
  • What security model governs the browser?

• Attacks Against Clients
  • Cross Site Scripting (XSS) and Response Splitting
  • Cross Site Request Forgery (CSRF)
  • Clickjacking

• Attacks Against Servers
  • SQL Injection
  • Unrestricted Uploads
  • CGI shell injection
Overview: The Web Model

• What is the web?
• What components make up today’s browsers and web servers?
• How has this functionality evolved over time?
• What security model governs the web browser?
What is the web?

• **Web (World Wide Web)**: A collection of data and services
  • Data and services are provided by **web servers**
  • Data and services are accessed using **web browsers** (e.g. Chrome, Firefox)
• The web is not the Internet
  • The Internet describes *how* data is transported between servers and browsers
Elements of the Web

• URLs: How do we uniquely identify a piece of data on the web?
• HTTP: How do web browsers communicate with web servers?
• Data on the webpage can contain:
  • HTML: A markup language for static webpages
  • CSS: A style sheet language for defining the appearance of webpages
  • Javascript: a programming language for running code in the web browser
Elements of the Web

• URLs: How do we uniquely identify a piece of data on the web?
• HTTP: How do web browsers communicate with web servers?

• Data on the webpage can contain:
  • HTML: A markup language for static webpages
  • CSS: A style sheet language for defining the appearance of webpages
  • Javascript: a programming language for running code in the web browser
Interacting with web servers

http://www.cs.swarthmore.edu/~chaganti/index.html

Protocol: ftp https tor

Hostname/server

• translated to an IP address by DNS

Path to the resource

index.html is static content
i.e., a fixed file returned by the server
Interacting with web servers

http://www.cs.swarthmore.edu/~chaganti/index.html

Protocol:  
- ftp  
- https  
- tor

Hostname/server translated to an IP address by DNS

Path to the resource
index.html is static content  
i.e., a fixed file returned by the server

http://facebook.com/delete.php

Path to the resource
delete.php is dynamic content  
i.e., a server generates the content on the fly
Interacting with web servers: dynamic content

http://facebook.com/delete.php   Path to the resource


server generates the content on the fly
URL Escaping


• URLs are designed to contain printable, human-readable characters (ASCII)
  • include non-printable characters in the URL?
• URLs have special characters that have assigned meaning (?, #, /)
• What if we want to use a special character in the URL?
  • Solution: URL encoding
  • Notation: Percent sign (%) followed by the hexadecimal value of the character
  • Example: %20 = ' ' (spacebar) %35 = '#' (hash sign)
    %50 = '2' (printable characters can be encoded too!)
• Security issues: makes scanning for malicious URLs harder
  • Suppose you want to block all requests to the path /etc/passwd
  • What if an attacker makes a request to %2F%65%74%63%2F%70%61%73%73%77%64?
HTTP: Hypertext transfer protocol

- client/server model
  - **client**: browser that requests, receives, (using HTTP protocol) and “displays” Web objects
  - **server**: Web server sends objects in response to requests

PC running Firefox browser

server running Apache Web server

iPhone running Safari browser
HTTP and the Web

First, a review…

- **web page consists of objects**
- object can be HTML file, JPEG image, Java applet, audio file,…
- web page consists of **base HTML-file** which includes several referenced objects
- each object is addressable by a **URL**, e.g.,

  \[ \text{www.someschool.edu/someDept/pic.gif} \]

  \[ \text{host name} \quad \text{path name} \]
HTTP Overview

1. User types in a URL.
   http://some.host.name.tld/directory/name/file.ext
   
   host name   path name
HTTP Overview

2. Browser establishes connection with server.
   Looks up “some.host.name.tld”
   connects //more on this later
HTTP Overview

3. Browser requests the corresponding data.
   GET /directory/name/file.ext HTTP/1.0
   Host: some.host.name.tld
   [other optional fields, for example:]
   User-agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
   Accept-language: en
HTTP Overview

4. Server responds with the requested data.
   HTTP/1.0 200 OK
   Content-Type: text/html
   Content-Length: 1299
   Date: Sun, 01 Sep 2013 21:26:38 GMT
   [Blank line]
   (Data data data data data...)
HTTP Overview

5. Browser renders the response, fetches any additional objects, and closes the connection.
Example

GET / HTTP/1.1
Host: demo.cs.swarthmore.edu

HTTP/1.1 200 OK
Vary: Accept-Encoding
Content-Type: text/html
Accept-Ranges: bytes
ETag: "316912886"
Last-Modified: Wed, 04 Jan 2017 17:47:31 GMT
Content-Length: 1062
Date: Wed, 05 Sep 2018 17:27:34 GMT
Server: lighttpd/1.4.35

Response Body
Example

GET / HTTP/1.1
Host: demo.cs.swarthmore.edu

Response Headers

<html><head><title>Demo Server</title></head>
<body>
.....
</body>
</html>
State(less)

(XKCD #869, “Server Attention Span”)
State(less)

- Original web: simple document retrieval
- Maintain State? Server is not required to keep state between connections
  
  ...often it might want to though

- Authentication: Client is not required to identify itself
  
  • server might refuse to talk otherwise though
- Server stores state, indexes it with a cookie
- Send this cookie to the client
- Client stores the cookie and returns it with subsequent queries to that same server
HTTP Request Header

GET / HTTP/1.1
Host: www.reddit.com
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,...
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9_2)...
DNT: 1
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US, en;q=0.8
Cookie: reddit_session=...
HTTP Response Header

HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Cache-Control: no-cache
Pragma: no-cache
x-frame-options: SAMEORIGIN
x-content-type-options: nosniff
x-xss-protection: 1; mode=block
Vary: accept-encoding
Content-Encoding: gzip
Content-Length: 24824
Server: '; DROP TABLE servertypes; --
Date: Mon, 10 Mar 2014 22:44:23 GMT
Connection: keep-alive
Cookies are key-value pairs

Set-Cookie:key=value; options; ....
Cookies are key-value pairs

Set-Cookie: key=value; options; ....

HTTP/1.1 200 OK
Date: Tue, 18 Feb 2014 08:20:34 GMT
Server: Apache

Set-Cookie: session-zdnet-production=6bhqcal1o0cbciagu11sisac2p3; path=/; domain=zdnet.com
Set-Cookie: zdregion=MTI5LjJuMTI5LjE1Mzplczp1czpjZDjmNlWY5TdkO1U1N2Q2Y2z3M5NGU3M2Y1ZTRmNG
Set-Cookie: zdregion=MTI5LjJuMTI5LjE1Mzplczp1czpjZDjmNlWY5TdkO1U1N2Q2Y2z3M5NGU3M2Y1ZTRmNG
Set-Cookie: edition=us; expires=Wed, 18-Feb-2015 08:20:34 GMT; path=/; domain=zdnet.com
Set-Cookie: session-zdnet-production=590b97fmrnqe4gb6vde4dvvq11; path=/; domain=zdnet.com
Set-Cookie: user_agent=desktop
Set-Cookie: zdnet_ad_session=f
Set-Cookie: firstpg=0
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-UA-Compatible: IE=edge,chrome=1
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 18922
Keep-Alive: timeout=70, max=146
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<html> ...... </html>
Cookies

Set-Cookie: edition=us; expires=Wed, 18-Feb-2015 08:20:34 GMT; path=/; domain=.zdnet.com

Client

Browser

(Private) Data

Semantics

- Store “us” under the key “edition” (think of it like one big hash table)
Cookies

Set-Cookie: edition=us; expires=Wed, 18-Feb-2015 08:20:34 GMT; path=/; domain=.zdnet.com

**Semantics**

- Store "us" under the key "edition" (think of it like one big hash table)
- This value is no good as of Wed Feb 18...
Cookies

```
Set-Cookie: edition=us; expires=Wed, 18-Feb-2015 08:20:34 GMT; path=/; domain=.zdnet.com
```

**Client**

- **Browser**
- **(Private) Data**

**Semantics**

- Store “us” under the key “edition” (think of it like one big hash table)
- This value is no good as of Wed Feb 18...
- This value should only be readable by any domain ending in `.zdnet.com`
Cookies

Set-Cookie: edition=us; expires=Wed, 18-Feb-2015 08:20:34 GMT; path=/; domain=.zdnet.com

**Semantics**

- Store “us” under the key “edition” (think of it like one big hash table)
- This value is no good as of Wed Feb 18...
- This value should only be readable by any domain ending in .zdnet.com
- This should be available to any resource within a subdirectory of /
Cookies

Semantics

- Store “us” under the key “edition” (think of it like one big hash table)
- This value is no good as of Wed Feb 18...
- This value should only be readable by any domain ending in .zdnet.com
- This should be available to any resource within a subdirectory of /
- Send the cookie to any future requests to <domain>/<path>
Cookies: keeping “state” (cont.)

Client:
- eBay 8734
- Cookie file

Server:
- Amazon server creates ID 1678 for user
- cookie: 1678

Backend database:
- Access
- cookie-specific action

One week later:
- eBay 8734
- Amazon 1678

Usual HTTP request msg:
- eBay 8734

Usual HTTP response set-cookie: 1678
- Amazon server creates ID 1678 for user
- cookie: 1678

Usual HTTP response msg:
- Access
- cookie-specific action

Usual HTTP request msg:
- Cookie: 1678
What Are Cookies Used For?

• Authentication
  • The cookie proves to the website that the client previously authenticated correctly

• Personalization
  • Helps the website recognize the user from a previous visit

• Tracking
  • Follow the user from site to site;
  • Read about iPads on CNN and see ads on Amazon 😱
  • How can an advertiser (A) know what you did on another site (S)?
HTTP Request/Responses with Cookies

HTTP/1.1 200 OK
Date: Tue, 18 Feb 2014 08:20:34 GMT
Server: Apache
Set-Cookie: session-zdnet-production=6bhqca1i0cbciagu11sisac2p3; path=/; domain=zdnet.com
Set-Cookie: zregion=MTi5LjiuMTi5LjE1Mzp1czp1czpjZDJmNWY5YTdkODU1N2Q2YzM5NGU3M2Y1ZTRmN6
Set-Cookie: zregion=MTi5LjiuMTi5LjE1Mzp1czp1czpjZDJmNWY5YTdkODU1N2Q2YzM5NGU3M2Y1ZTRmN6
Set-Cookie: edition=us; expires=Wed, 18-Feb-2015 08:20:34 GMT; path=/; domain=zdnet.com
Set-Cookie: session-zdnet-production=59ob97fpinqe4bg6lde4dvvq11; path=/; domain=zdnet.com

Subsequent visit

HTTP Headers
http://zdnet.com/
GET / HTTP/1.1
Host: zdnet.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.11) Gecko/201001013 Ubuntu/9.04 (jaunty) Firefox/3.6.11
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Cookie: session-zdnet-production=59ob97fpinqe4bg6lde4dvvq11 zregion=MTi5LjiuMTi5LjE1Mzp1czp1czpjZDJmNW
Cookies and Privacy

Cookies permit sites to learn a lot about you

- supply name and e-mail to sites (and more!)
- third-party cookies (ad networks) follow you across multiple sites.
Why use cookies?

- **Tracking users**
  - Advertisers want to know your behavior
  - Ideally build a profile *across different websites*
    - Read about iPad on CNN, then see ads on Amazon?!
  - How can an advertiser (A) know what you did on another site (S)?
Why use cookies?

• Tracking users
  • Advertisers want to know your behavior
  • Ideally build a profile across different websites
    - Read about iPad on CNN, then see ads on Amazon?!
  • How can an advertiser (A) know what you did on another site (S)?

S shows you an ad from A; A scrapes the referrer URL
Why use cookies?

- Tracking users
  - Advertisers want to know your behavior
  - Ideally build a profile across different websites
    - Read about iPad on CNN, then see ads on Amazon?!
  - How can an advertiser (A) know what you did on another site (S)?

S shows you an ad from A; A scrapes the referrer URL

Option 1: A maintains a DB, indexed by your IP address

Problem: IP addrs change
Why use cookies?

- **Tracking users**
  - Advertisers want to know your behavior
  - Ideally build a profile *across different websites*
    - Read about iPad on CNN, then see ads on Amazon?!
  - How can an advertiser (A) know what you did on another site (S)?

<table>
<thead>
<tr>
<th>Option 1: A maintains a DB, indexed by your IP address</th>
<th>Problem: IP addr change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option 2: A maintains a DB indexed by a <em>cookie</em></td>
<td>- “Third-party cookie”</td>
</tr>
<tr>
<td></td>
<td>- Commonly used by large ad networks (doubleclick)</td>
</tr>
</tbody>
</table>
Cookie tracking
Cookie tracking
Cookie tracking

Snippet of reddit.com source
Cookie tracking

Snippet of reddit.com source

Our first time accessing adzerk.net
Cookie tracking

I visit reddit.com

Later, I go to reddit.com/r/security
Cookie tracking

I visit reddit.com

HTTP Headers

GET /reddit/ads.html?sr=-reddit.com,loggedout&bust2=http://www.reddit.com HTTP/1.1
Host: static.adzerk.net
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.11) Gecko/20100113 Ubuntu/9.04 (jaunty) Firefox/3.6.11
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8; q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://www.reddit.com/

HTTP/1.1 200 OK
Date: Thu, 19 Feb 2015 17:37:51 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: __cfuid=d3ca93cd30ca47b76600d63cd283e9b81424367471; expires=Fri, 19-Feb-16 17:37:51 GMT; path=/; domain=.adzerk.net...

We are only sharing this cookie with *.adzerk.net; but we are telling them about where we just came from

Later, I go to reddit.com/r/security

HTTP Headers

GET /reddit/ads.html?sr=security,loggedout&bust2=http://www.reddit.com HTTP/1.1
Host: static.adzerk.net
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.11) Gecko/20100113 Ubuntu/9.04 (jaunty) Firefox/3.6.11
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8; q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://www.reddit.com/r/security
Cookie: __cfduid=d3ca93cd30ca47b76600d63cd283e9b81424367471

I visit reddit.com
Cookies and Privacy

Cookies permit sites to learn a lot about you

You could turn them off ...but good luck doing anything on the internet!
Cookies and web authentication

• An extremely common use of cookies is to track users who have already authenticated

• If the user already visited http://website.com/login.html?user=alice&pass=secret with the correct password, then the server associates a “session cookie” with the logged-in user’s info

• Subsequent requests (GET and POST) include the cookie in the request headers and/or as one of the fields: http://website.com/doStuff.html?sid=81asf98as8eak

• The idea is for the server to be able to say “I am talking to the same browser that authenticated Alice earlier.”