Off topic

- **Nerd Night DC**
  - Saturday, February 20, 2010
  - 6:00pm - 10:00pm
- DC9, 1940 9th St NW, Washington, DC
- Topics: Maggots, Calvin and Hobbes, Stormchasing
Project 3

- Distributed to your linuxlab CVS repository
- projects/pictureParser
- check it out
- update your course manager plugin
- Install ANTLR and ANTLR eclipse plugins
Get to work

- A couple of people only made significant progress on project 2 in the last 24 hours before the deadline, or even after the deadline
- Get started early
Grammar actions

- Adding Java code to your grammars
- view [Java Dude ANTLR tutorial, part 7](#)
Web Security

- http://cwe.mitre.org/top25/ - top 25 security errors
- OWASP wiki
  - http://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
  - http://www.owasp.org/index.php/CSRF
Web and web security Quiz

- Tuesday
- Sessions
- Linking form entries with database entries
- Security: SQL Injection, XSS, CSRF, session hijacking
Trusted data

• Anything coming from the browser shouldn't be trusted

• Including parameters for which the HTML should limit the possible choices
Different kinds of escaping needed for different contexts

• The rules about what needs to be escaped are complex, and depend upon how the value will be used

• [http://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet](http://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet)

• Bad guys are inventive

• Generally, prefer white listing rather than black listing

• what is guaranteed safe, rather than what is known bad
Bad data from malicious users

- Any data can be inspected/transformed
  - e.g., see WebScarab
  - hidden fields, headers, anything
- A user with a pwned machine is malicious
  - as is a user with a pwned web proxy
- A user viewing XSS is somewhat malicious
Bad data from good users

• Send a link in email, web page, whatever
  • will generate GET request
• Show html page with auto-submitting form
  • can generate GET or POST request
Watch error reports

We're Sorry

Your current session has expired. To start over, please return to united.com.

SessionId :1266069911740209087120174292170UserInt30
Session management

• Normally, actions are only performed on behalf of signed in users

• But sessions are often handled via cookies

• If the user is signed in, *any* request from that browser is considered to be signed in
Session Hijacking

- If a bad guy can get your session identifier, they can send it to their own server
- and they can hijack your session
Preventing session hijacking

• Store client IP address for session; if session changes to a different IP address, must be a session hijack, right?
IP address

- IP addresses change
  - Most commonly, moving between WiFi network and 3G network
  - plus, can happen with DHCP
- Might be hijacked to different machine with same IP address
  - both requests via same NAT box
How to steal a session

• Session fixation
  • start a session, get a session id (not yet logged in), get user to follow continue with that session id
  • fixed by always assigning new session id when logging in

• Prediction

• Sniffing

• Javascript (typically due to XSS)
Javascript cookie stealing

- Somewhat preventing by HTTP only cookies
- Only sent to server, not visible to client side javascript
- Not fully supported on all browsers
Cross site request forgery

- If you don't have any XSS vulnerabilities
- CSRF should prevent/limit bad data from good users
- Only act on requests that our website generated
CRSF

• Sometimes, you want to allow other sites to generate requests to your web site

• for example, "show tickets for Black Eyed Peas"

• But requests to change password or transfer funds…

• only we should generate such requests and perform them for signed in users
Can check referrer field

- From good users, if referrer header present, generally trusted
- If you get a POST request and the referrer field doesn't match, reject the
What if referrer field not present?

- If you want to handle this situation, then you need to include a secret in every link/form.
- can use a hidden form field, or encode it directly in the URL.