Intro to Twitter Hacking

Review of Twitter, some code from Chapter 1

Outline

- What is Twitter?
  - Brief twitter conventions
- Sample Twitter Hacking (code)

What is Twitter?

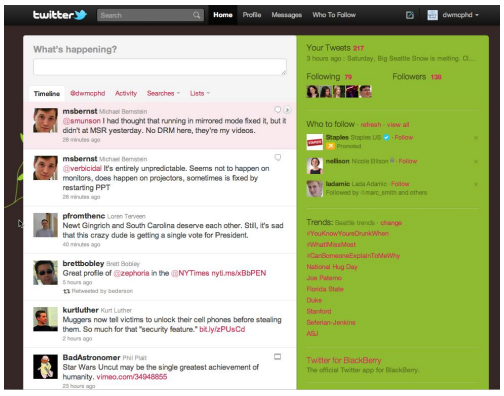
What is Twitter?

- Micro blog
  - Status messages
  - 140 characters max
  - Links

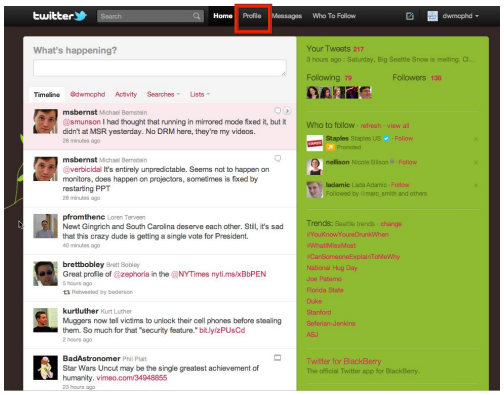
What is Twitter?

- Micro blog
  - Status messages
  - 140 characters max
  - Links
- Social
  - Follow (be followed)
  - Groups

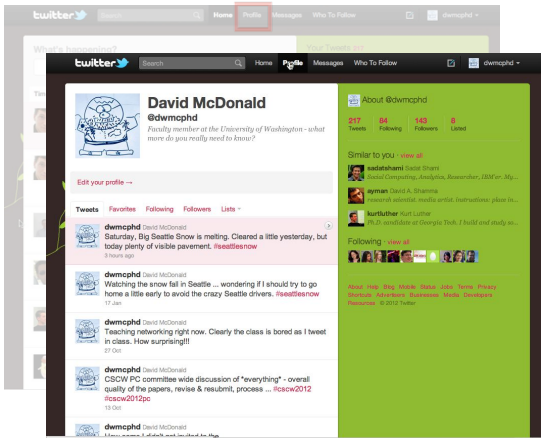
What is Twitter?



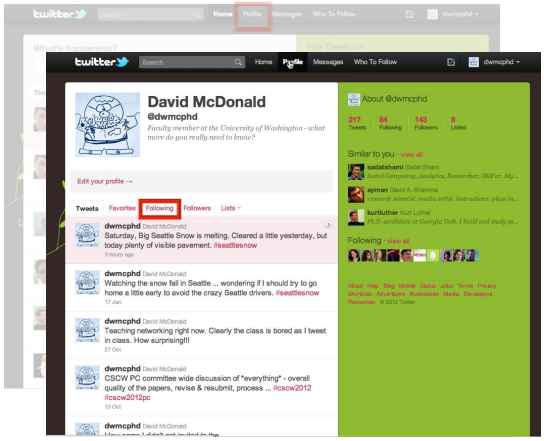
What is Twitter?



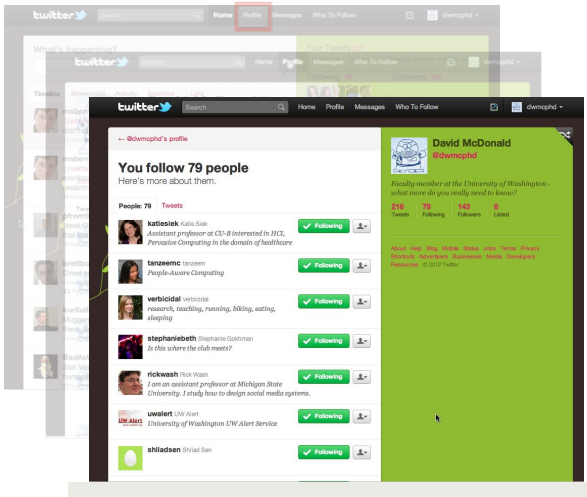
What is Twitter?



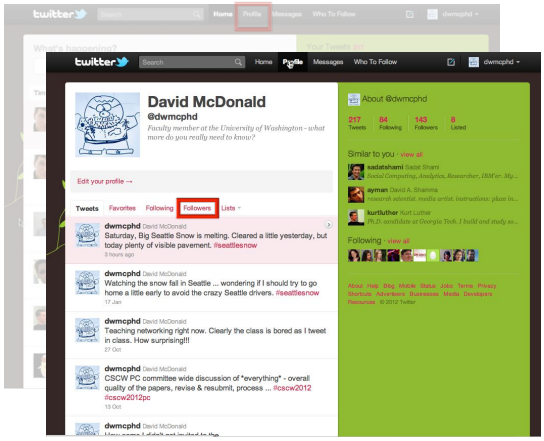
What is Twitter?



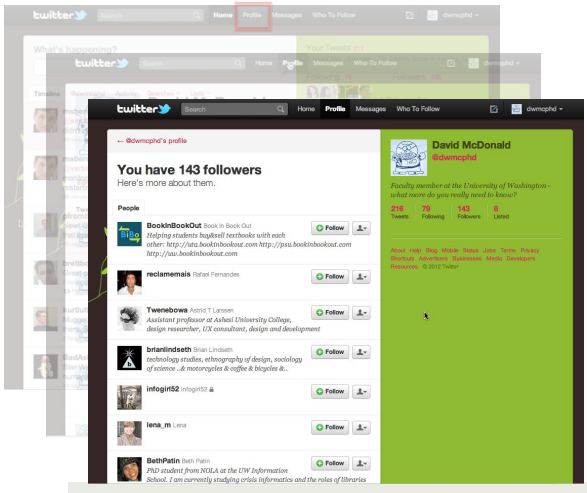
What is Twitter?



What is Twitter?



What is Twitter?



Twitter Conventions

- Directed Tweet
  - @username
- Re-tweet
  - RT, @username (or) rt @username
  - via @username
- Categorization, tagging
  - #category

Sample Code – collecting tweets

```
import cPickle, nltk, twitter
t_search = twitter.Twitter(domain="search.twitter.com")
madonna_results = []
for page in range(1,6):
       madonna_results.append(t_search.search(q="Madonna", rpp=100, page=page))
mtweets = [ tweet
       for page in madonna_results
                  for tweet in page['results'] ]
mttext= [ r['text']
    for result in madonna_results
        for r in result['results'] ]
mwords = []
for t in mttext:
       mwords += [ w for w in t.split() ]
print "Number of words in Madonna related tweets:",len(mwords)
```

Sample Code – pickling tweets

```
## Assuming prior imports and code

f = open("versus.madonna.tweets.pickle","wb")
cPickle.dump(mtweets,f)
f.close()
f = open("versus.madonna.ttext.pickle","wb")
cPickle.dump(mttext,f)
f.close()
f = open("versus.madonna.words.pickle","wb")
cPickle.dump(mwords,f)
f.close()
```

Sample Code – building graph

```
import re, cPickle, nltk, twitter
import networkx as nx
rt_patterns = re.compile(r"(RT|via)((?:\b\W*@\w+)+)", re.IGNORECASE)
mg = nx.DiGraph()
mtweets = cPickle.load(open("versus.madonna.tweets.pickle"))
def get_rt_sources(tweets):
    return [ source.strip()
                        for tuple in rt_patterns.findall(tweets)
                            for source in tuple
                                if source not in ("RT", "via") ]
def process_tweets(tweets, graph):
    for t in tweets:
        rt_sources = get_rt_sources(t["text"])
        if not rt_sources:
            continue
        for rt_source in rt_sources:
            graph.add_edge(rt_source, t["from_user"], {"tweet_id": t["id"]})
```

Sample Code – Saving graph

```
def write_dot_file(graph,fname):
    try:
        nx.drawing.write_dot(graph,fname)
    except ImportError, e:
        dot = ['"%s" -> "%s" [tweet_id=%s]' % (n1, n2, graph[n1][n2]['tweet_id'])
                        for n1,n2 in graph.edges()]
        dotenc = [ item.encode('ascii','replace')
                            for item in dot ]
        f = open(fname,"w")
        f.write('strict digraph {\n%s\n}'%(';\n'.join(dotenc),))
        f.close()
```

Run Sample

- I reworked the sample
- Golden Globes – Madonna versus Elton John controversy
- Saving intermediary values
- Build diGraph, visualize