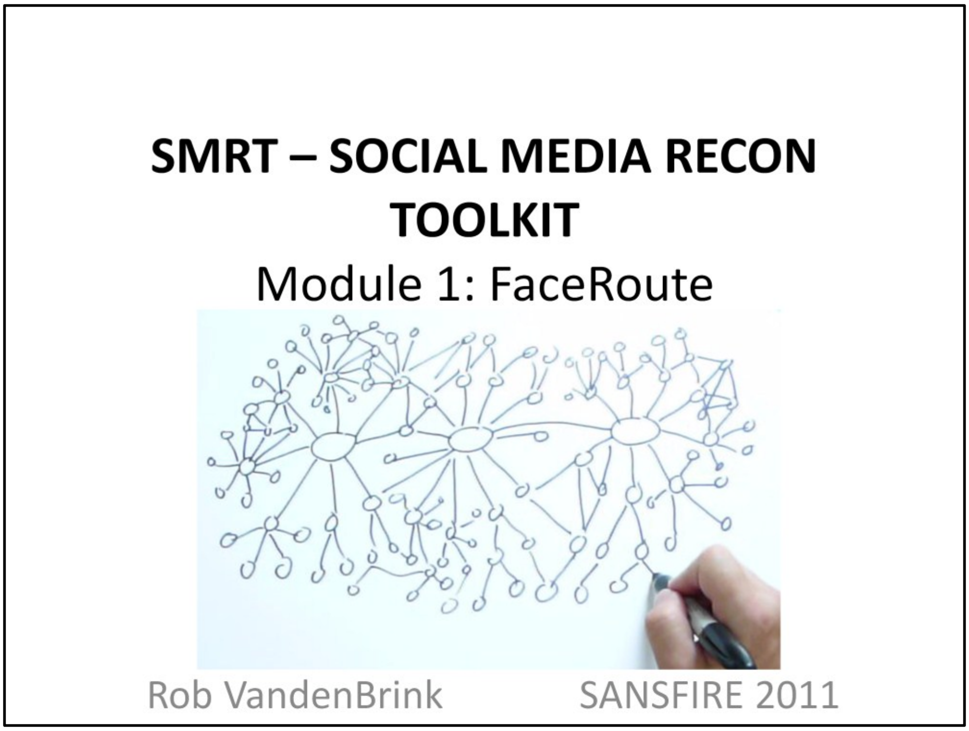
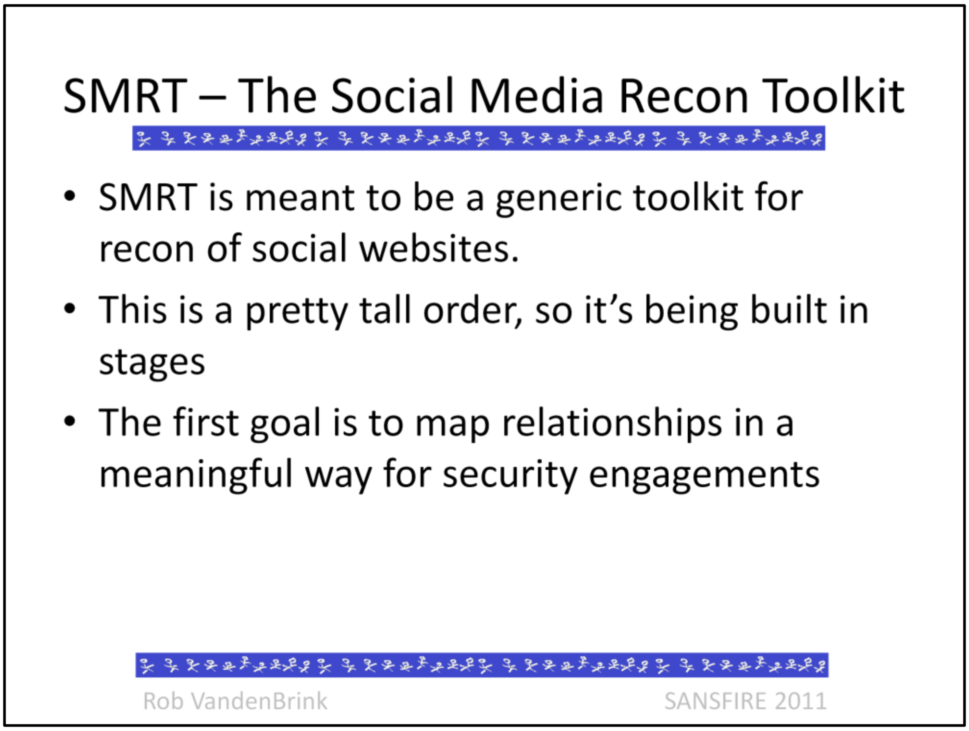


SMRT – SOCIAL MEDIA RECON TOOLKIT

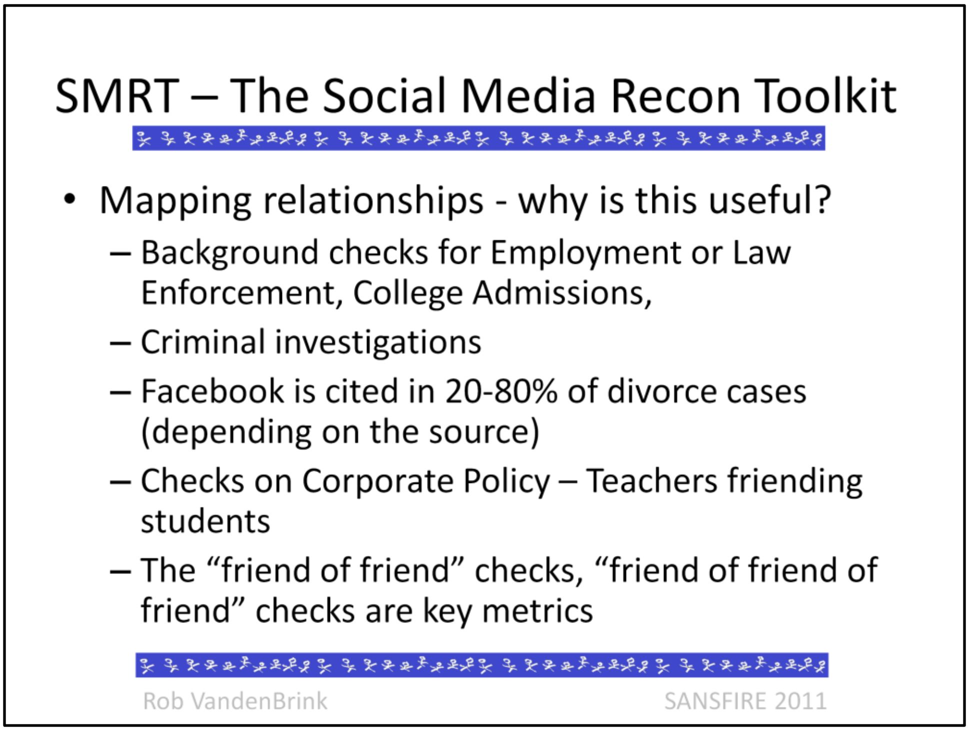
Module 1: FaceRoute

Rob VandenBrink SANSFIRE 2011



SMRT – The Social Media Recon Toolkit

- SMRT is meant to be a generic toolkit for recon of social websites.
- This is a pretty tall order, so it’s being built in stages
- The first goal is to map relationships in a meaningful way for security engagements



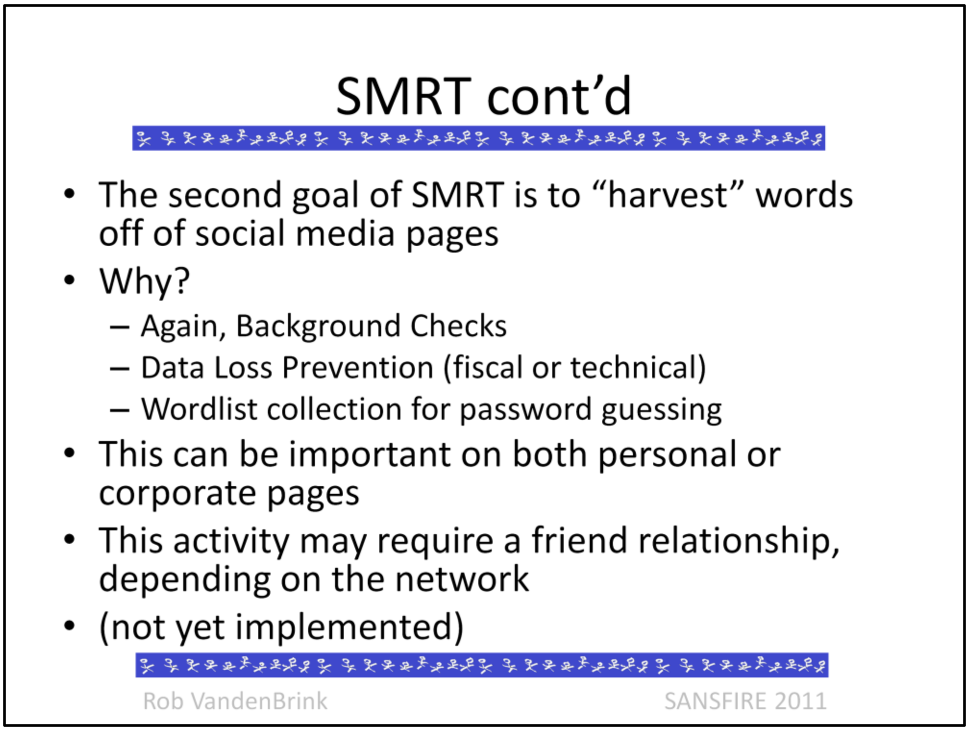
SMRT – The Social Media Recon Toolkit

- Mapping relationships - why is this useful?
  - Background checks for Employment or Law Enforcement, College Admissions,
  - Criminal investigations
  - Facebook is cited in 20-80% of divorce cases (depending on the source)
  - Checks on Corporate Policy – Teachers friending students
  - The “friend of friend” checks, “friend of friend of friend” checks are key metrics

Rob VandenBrink SANSFIRE 2011

SMRT is a toolset that has a few purposes. The initial and main purpose (which we’ll discuss today) is to map relationships. Why might this be useful?

- The use that immediately comes to mind is background checks for employment. Employers routinely use Facebook, Twitter and Linkedin as an informal metric to measure judgment. For instance, if you discuss previous employers in a disparaging manner, that would probably count against you, as would discussing illegal activity or posting inappropriate photos or content.
- Law enforcement will similarly use social media in background checks. It’s also common to see law enforcement `are used in something like 20-80% of divorce cases.
- Auditing personal activity against corporate policy is a growing trend in social media. For instance, some school boards mandate that teachers do NOT “friend” students on Facebook and other sites, while other School Boards encourage online social interaction. The FACEROUTE package we’ll discuss today has been used to assess Teacher online behaviour in both cases.
- In many cases, a basic friend list is all that’s considered (the school board example for instance). In this case, you’d think you could just use a browser to display a friend list. However, in assessing a school with 75 teachers and over 1,000 students, FACEROUTE is useful in both dealing with the volume and flagging audit findings.
- In many other cases, however, the “friend of a friend” list is also important. Background checks of all kinds for instance will routinely collect this information. In a social world where friend lists are routinely over 200 (who has 200 real friends?), a single “friend of a friend” check can easily surpass 40,000 nodes across 200 pages. This check is simply not manageable using manual methods.



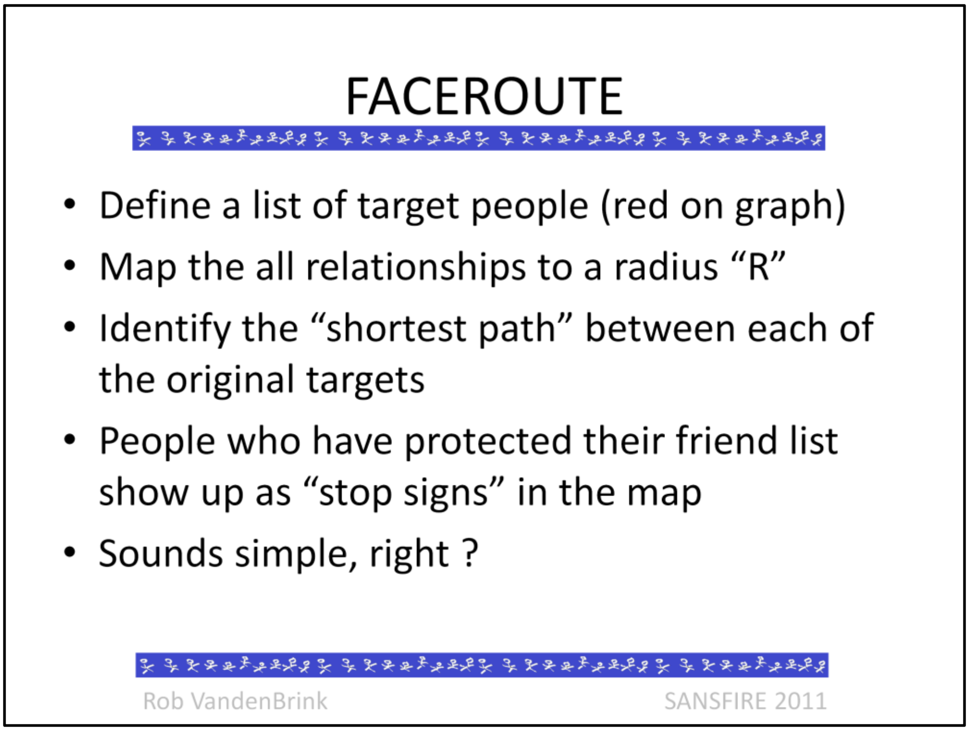
SMRT cont’d

- The second goal of SMRT is to “harvest” words off of social media pages
- Why?
  - Again, Background Checks
  - Data Loss Prevention (fiscal or technical)
  - Wordlist collection for password guessing
- This can be important on both personal or corporate pages
- This activity may require a friend relationship, depending on the network
- (not yet implemented)

Rob VandenBrink SANSFIRE 2011

Along the way, it was realized that dumping an entire personal site and creating a wordlist might also be useful.

- Wordlists of this kind can be used to check for data leakage of intellectual property or other confidential information (grepping such a list against a list of “words of interest” for instance).
- This can be used against both personal and corporate pages on social sites – for instance, it’s often policy to audit marketing communications to ensure that new products are not pre-announced.
- Wordlists from social sites can also be used as input for password guessing lists (names, events and hobbies for instance). In today’s world of rainbow tables and advanced brute force methods (using FPGA’s and fast processors), you’d think that this use would not be so popular, but a variant of FACEROUTE that simply dumps information has proved useful in auditing passwords in a DLP context (if your password is on your Facebook or Linkedin page, you are essentially posting your password)



FACEROUTE

- Define a list of target people (red on graph)
- Map the all relationships to a radius “R”
- Identify the “shortest path” between each of the original targets
- People who have protected their friend list show up as “stop signs” in the map
- Sounds simple, right ?

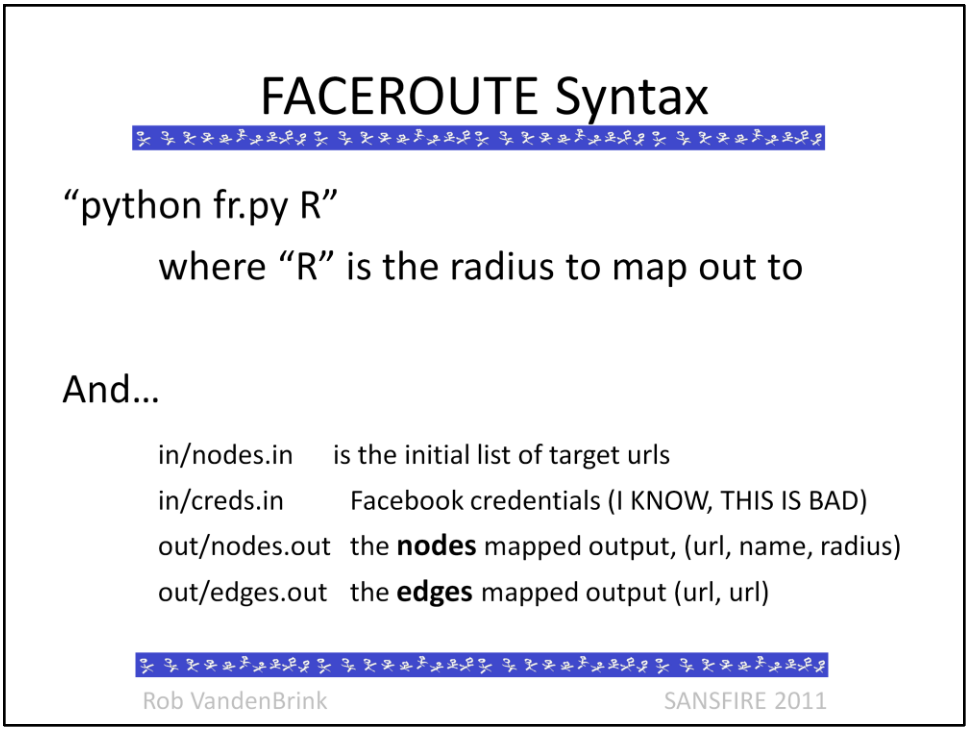
Rob VandenBrink SANSFIRE 2011

Faceroute is the first module in the SMRT toolset

Besides basic mapping, what should it do?

- Maps the original list of target people in red
- Maps the “SPF” (Shortest Path First) route between all of the orginal target people in red
- All other nodes are in light grey for readability
- A “stop sign” node indicates people who have gone to the trouble to set their privacy settings in Facebook such that friends cannot be listed

Believe me, this is NOT as simple as it sounds.



FACEROUTE Syntax

“python fr.py R”

where “R” is the radius to map out to

And…

in/nodes.in is the initial list of target urls

in/creds.in Facebook credentials (I KNOW, THIS IS BAD)

out/nodes.out the **nodes** mapped output, (url, name, radius)

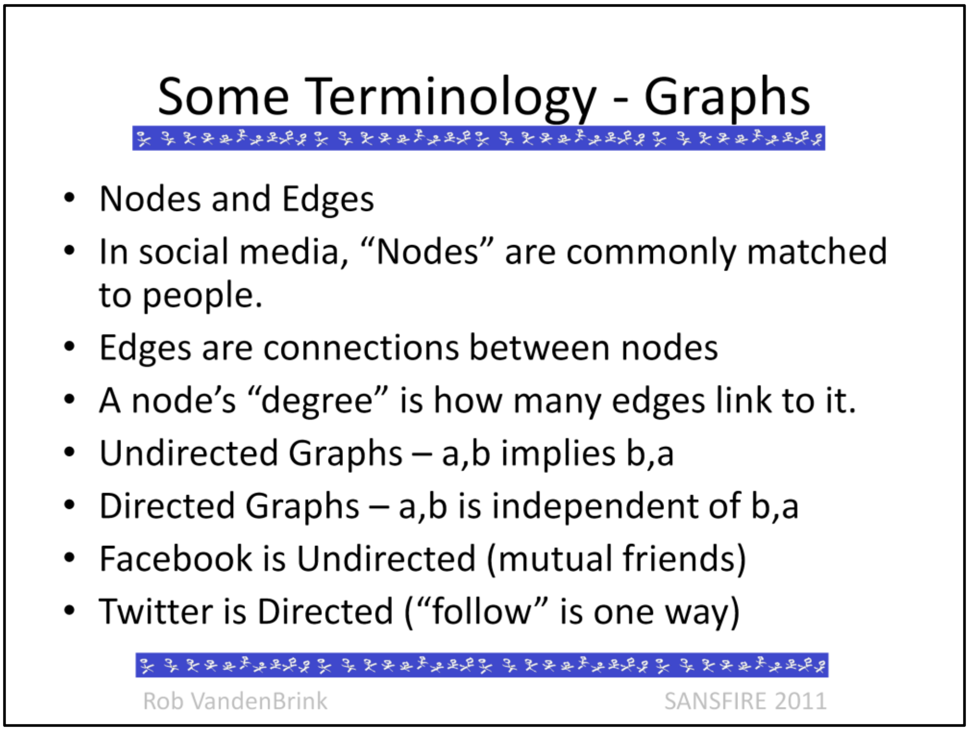
out/edges.out the **edges** mapped output (url, url)

Rob VandenBrink SANSFIRE 2011

Faceroute is the first module in the SMRT toolset

Besides basic mapping, what does it do?

- Maps the original list of target people in red
- Maps the “SPF” (Shortest Path First) route between all of the orginal target people in red
- All other nodes are in light grey for readability
- A “stop sign” node indicates people who have gone to the trouble to set their privacy settings in Facebook such that friends cannot be listed



Some Terminology - Graphs

- Nodes and Edges
- In social media, “Nodes” are commonly matched to people.
- Edges are connections between nodes
- A node’s “degree” is how many edges link to it.
- Undirected Graphs – a,b implies b,a
- Directed Graphs – a,b is independent of b,a
- Facebook is Undirected (mutual friends)
- Twitter is Directed (“follow” is one way)

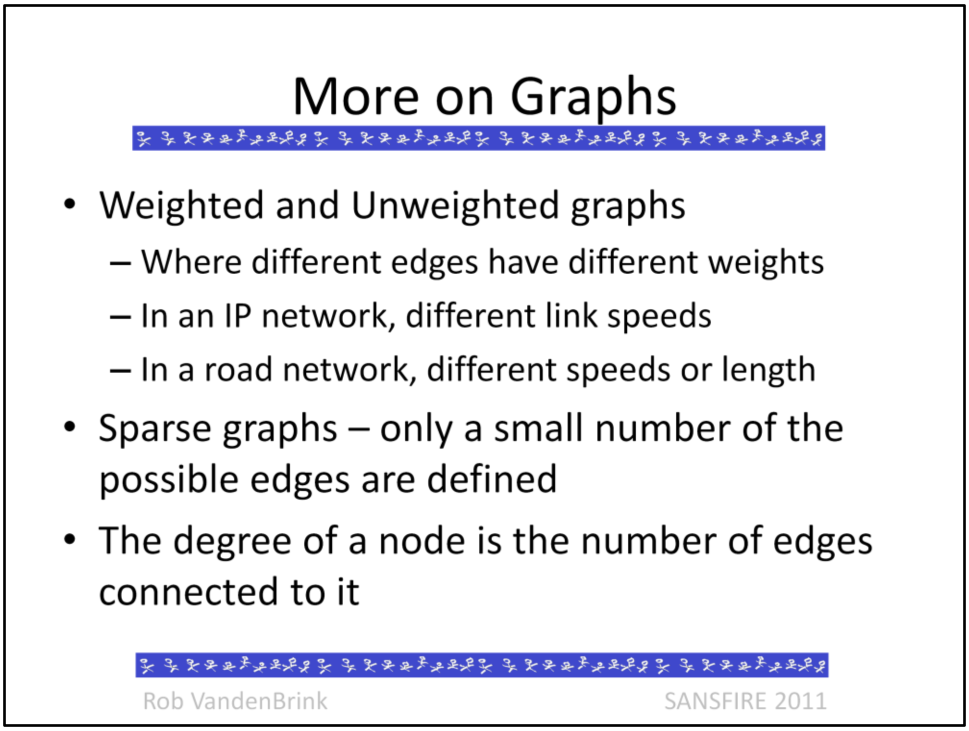
Rob VandenBrink SANSFIRE 2011

In the context of mapping, the word “graph” has a special meaning. In the case of mapping social media relationships, it implies a graph showing all interesting “nodes” (people) and “edges” (connection between nodes, or people in this case).

A nodes “degree” is a simple concept – it’s the sum of all edge values for a specific node. Or in our case, the count of connection that person has.

Edges can have direction. In the case of Facebook, edges are undirected, a link works both ways – if you have friended someone, they also have friended you. Twitter on the other hand is a directed network – you can follow someone, but they don’t necessarily follow you.

An analogy in real life might be useful here.
The set of “have shaken hands with” is definitely a undirected graph. Shaking hands with someone is a mutual experience. On the other hand, “have heard of” is undirected. I’ve certainly heard of Steven Hawking, but I’m pretty sure he hasn’t heard of me.



More on Graphs

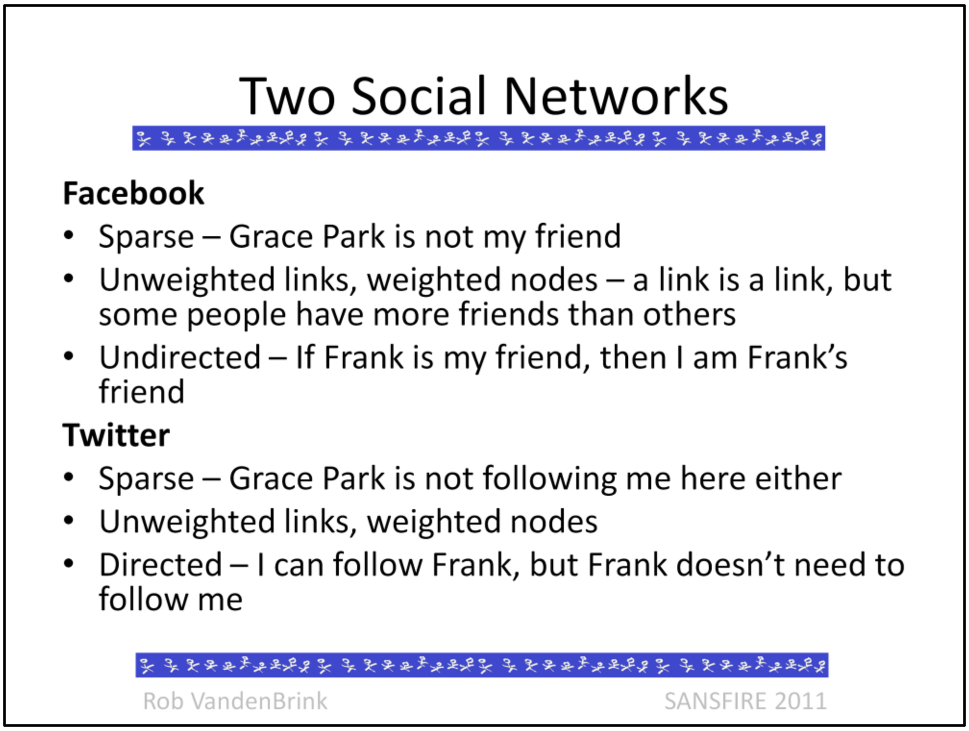
- Weighted and Unweighted graphs
  - Where different edges have different weights
  - In an IP network, different link speeds
  - In a road network, different speeds or length
- Sparse graphs – only a small number of the possible edges are defined
- The degree of a node is the number of edges connected to it

Rob VandenBrink SANSFIRE 2011

Those familiar with mapping networks will be familiar with the concept of “path cost” – in this context we would account for path cost by adding a value to edges, giving some connections a higher preference than others. For instance, in the case of a road network, a GPS might prefer paved roads over gravel roads, and expressways over both. The speed limit on a road, and the ‘drive length’ of any path will also help define the weight (by computing overall drive time). In the context of the SMRT and FACEROUTE toolset, we’ll assign a value of 1 to each edge. We’re not measuring BFF’s, we’re just mapping relationships.

Sparse graphs describe a system where the number of edges is low, compared to the number of nodes. In other words, there’s room in the graph for more edges between nodes. A city if a real-life example of a sparse system. To cite a very common example, I might live in Canada, but I don’t know everyone in Canada.

Dense graphs describe systems where the number of edges is high. In other words, most nodes connect to most other nodes. Most High Schools would be considered a dense system – most students in school know most other students.



Two Social Networks

Facebook

- Sparse – Grace Park is not my friend
- Unweighted links, weighted nodes – a link is a link, but some people have more friends than others
- Undirected – If Frank is my friend, then I am Frank's friend

Twitter

- Sparse – Grace Park is not following me here either
- Unweighted links, weighted nodes
- Directed – I can follow Frank, but Frank doesn't need to follow me

Rob VandenBrink SANSFIRE 2011

Facebook
Overall, Facebook is considered a sparse system. Sadly, even though Grace Park has a Facebook page, she is not my friend on Facebook.

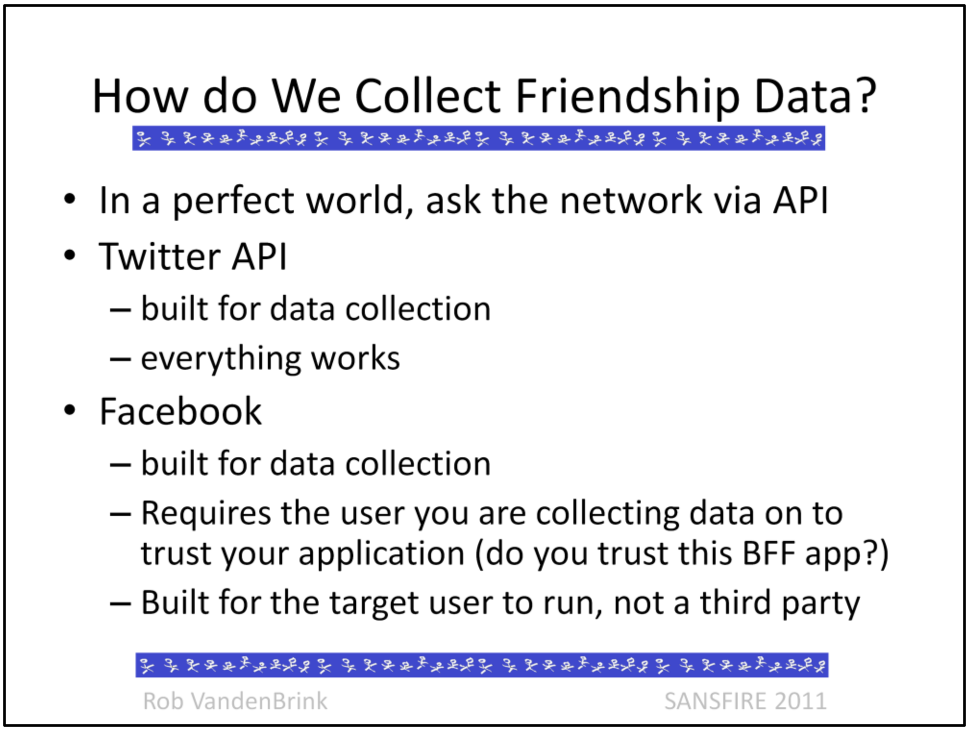
In most applications you would consider each link to be equal (a friend is a friend is a friend), but some people do have more friends than others, so some people have more weight than others, their degree measured by their friend count.

Facebook is an undirected system – if you friend someone, they also friend you.

Twitter
Twitter is similarly a sparse system. Grace Park doesn't follow me here either.

Links are directed however, as we noted earlier, I can follow Steven Hawking, but he's probably not following me.

Again, in most applications you wouldn't weight edges (a link is a link), but you can measure the degree of graph members – in Twitter you normally calculate degree by follower count, rather than the count of people you follow. The follower count is generally considered a measure of value of content posted. Though in the case of Justin Bieber that might not be the case ..



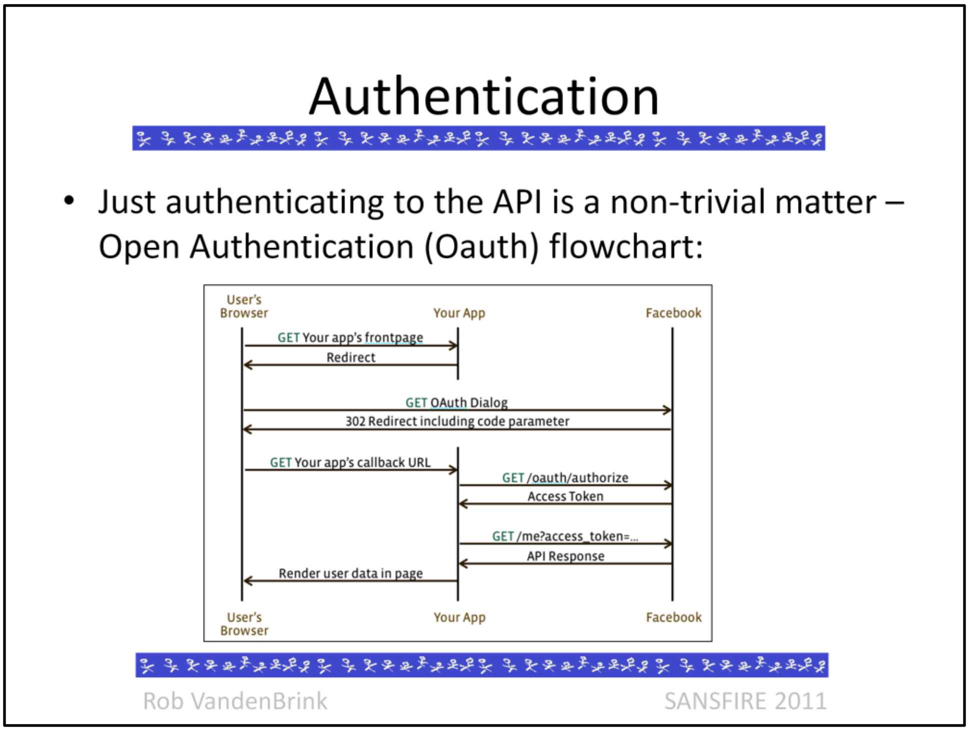
How do We Collect Friendship Data?

- In a perfect world, ask the network via API
- Twitter API
  - built for data collection
  - everything works
- Facebook
  - built for data collection
  - Requires the user you are collecting data on to trust your application (do you trust this BFF app?)
  - Built for the target user to run, not a third party

Rob VandenBrink SANSFIRE 2011

All of the more popular social sites now publish an API, and there are generally libraries available for most popular languages to simplify the use of these API's.

However "Simplify" is a relative term…



Authentication

- Just authenticating to the API is a non-trivial matter – Open Authentication (Oauth) flowchart:

User's Browser

Your App

Facebook

GET Your app's frontpage

Redirect

GET OAuth Dialog

302 Redirect including code parameter

GET Your app's callback URL

GET /oauth/authorize

Access Token

GET /me?access_token=...

API Response

Render user data in page

User's Browser

Your App

Facebook

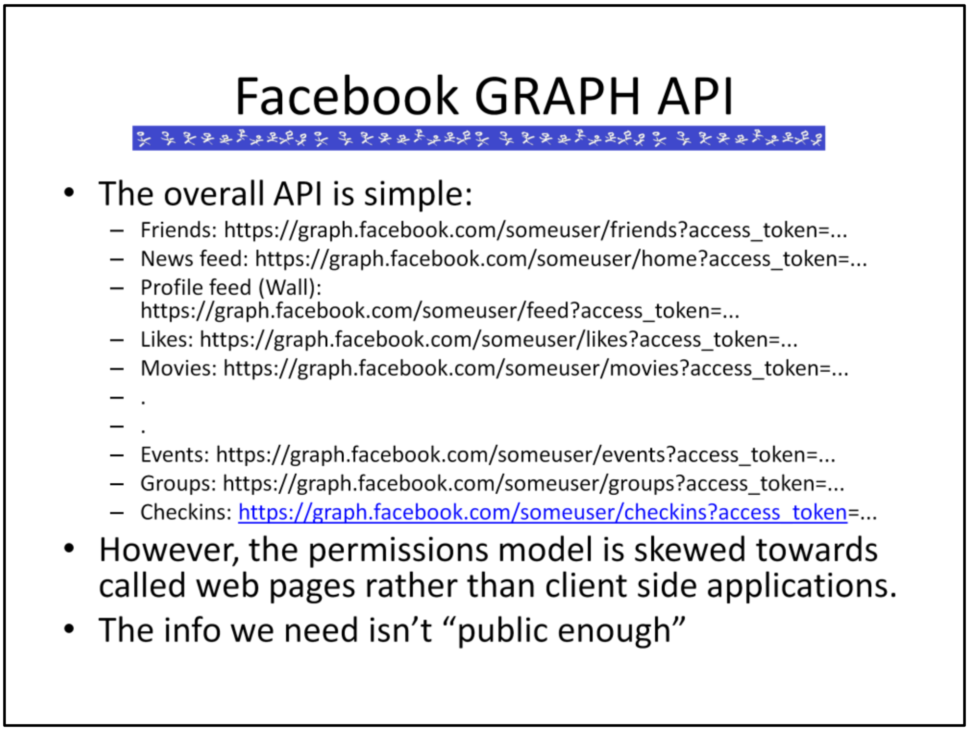
Rob VandenBrink

SANSFIRE 2011

Open Authentication (OAUTH) is how most of the social sites are steering their APIs – your app needs to authenticate to the site using OAUTH before any data is available to it.

Note that the exchange for authentication is *very* complex, with the end result being an “authentication token” which can be used to process individual transactions. This normally involves registering an application, with application credentials (normally a name and a “secret”), but certificates can also be used.

More on Oauth can be found at:
http://oauth.net/
http://developers.facebook.com/docs/authentication/

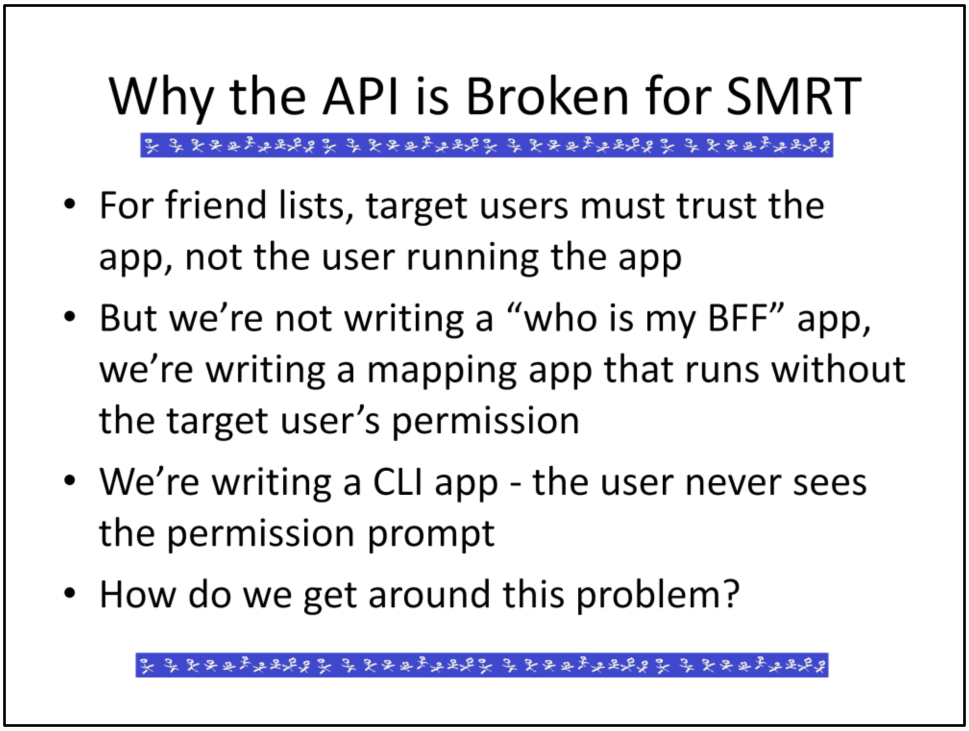


Facebook GRAPH API

- The overall API is simple:
  - Friends: https://graph.facebook.com/someuser/friends?access_token=...
  - News feed: https://graph.facebook.com/someuser/home?access_token=...
  - Profile feed (Wall): https://graph.facebook.com/someuser/feed?access_token=...
  - Likes: https://graph.facebook.com/someuser/likes?access_token=...
  - Movies: https://graph.facebook.com/someuser/movies?access_token=...
  - .
  - .
  - Events: https://graph.facebook.com/someuser/events?access_token=...
  - Groups: https://graph.facebook.com/someuser/groups?access_token=...
  - Checkins: https://graph.facebook.com/someuser/checkins?access_token=...
- However, the permissions model is skewed towards called web pages rather than client side applications.
- The info we need isn't "public enough"

Once authenticated, the API seems very simple – just browse to the appropriate link, and the data is returned.
(the access token is returned to the app as part of the authentication handshake)

Aside from the complexity of the authentication and access token process, this seems simple, right?



Why the API is Broken for SMRT

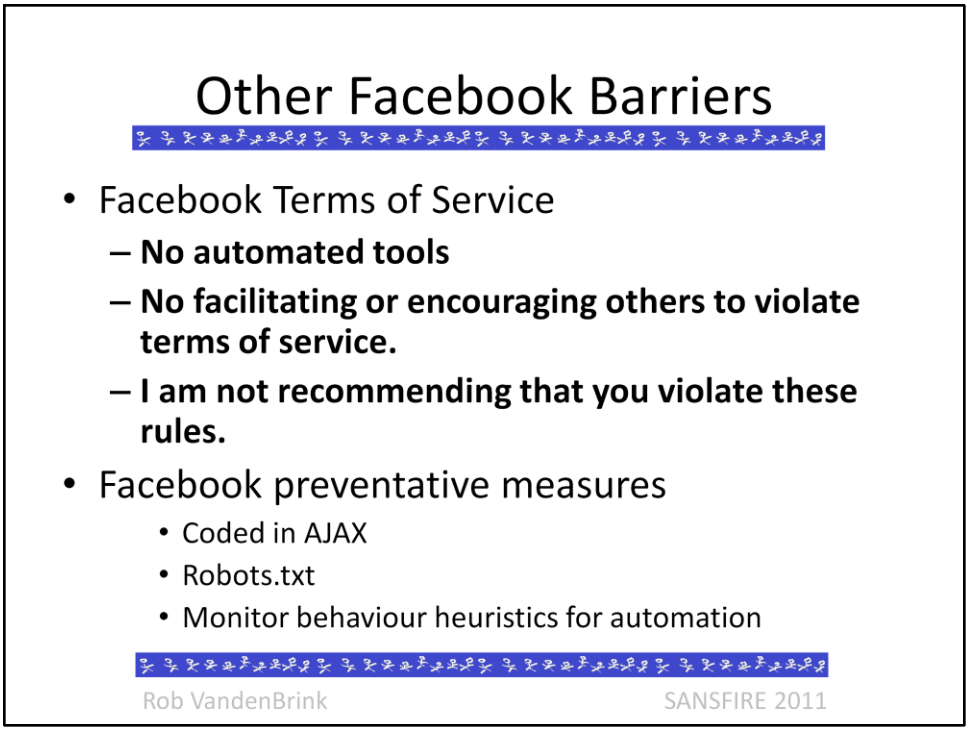
- For friend lists, target users must trust the app, not the user running the app
- But we’re not writing a “who is my BFF” app, we’re writing a mapping app that runs without the target user’s permission
- We’re writing a CLI app - the user never sees the permission prompt
- How do we get around this problem?

Who writes an API that has fewer rights than the native client (in this case, the browser).

Why, Facebook does !

Even if users could see the prompt for permission, a recon app like Faceroute can’t wait for days as each target gets around to logging in and responding, and (hopefully) the targets should say “no” anyway

All the while, there’s a perfectly good interface (any browser) that can collect the information we need.



Other Facebook Barriers

- Facebook Terms of Service
  - **No automated tools**
  - **No facilitating or encouraging others to violate terms of service.**
  - **I am not recommending that you violate these rules.**
- Facebook preventative measures
    - Coded in AJAX
    - Robots.txt
    - Monitor behaviour heuristics for automation

Note that Facebook includes in it's Terms of Service:

> ***"You will not collect users' content or information, or otherwise access Facebook, using automated***
> ***means (such as harvesting bots, robots, spiders, or scrapers) without our permission."***

Also:

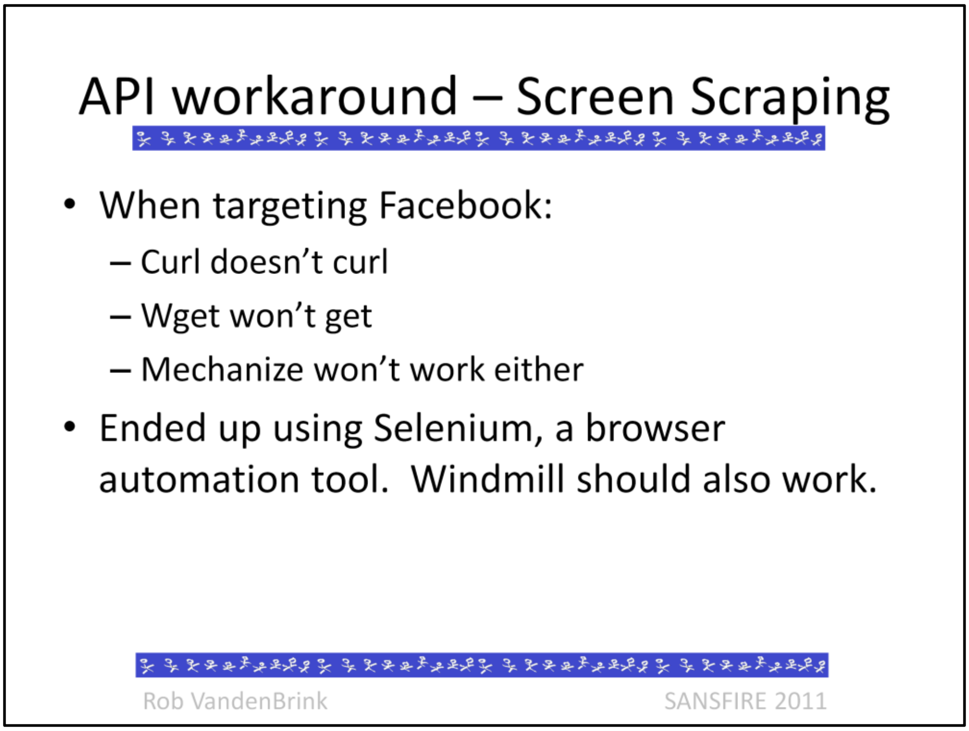
> ***"You will not facilitate or encourage any violations of this Statement."***

This pretty much precludes any meaningful recon on Facebook, and makes you wonder what that API is for ??
It also means that I am not encouraging you to use SMRT or FACEROUTE. In fact, it won't be posted until the graphical issues are worked out.

Along with terms of service, Facebook also aggressively uses ROBOTS.TXT, flagging pretty much everything as "don't look here"

Facebook also monitors browsing behaviour to detect automation, and implements CAPTCHAs to prevent it

The use of AJAX means that most of the site is dynamic in nature – pages are essentially created on the fly (you can see this especially in the friends list pages).



API workaround – Screen Scraping

- When targeting Facebook:
  - Curl doesn’t curl
  - Wget won’t get
  - Mechanize won’t work either
- Ended up using Selenium, a browser automation tool. Windmill should also work.

Rob VandenBrink SANSFIRE 2011

Because of the dynamic nature of the site, you cannot, for instance, use wget or curl with a target url and credentials.

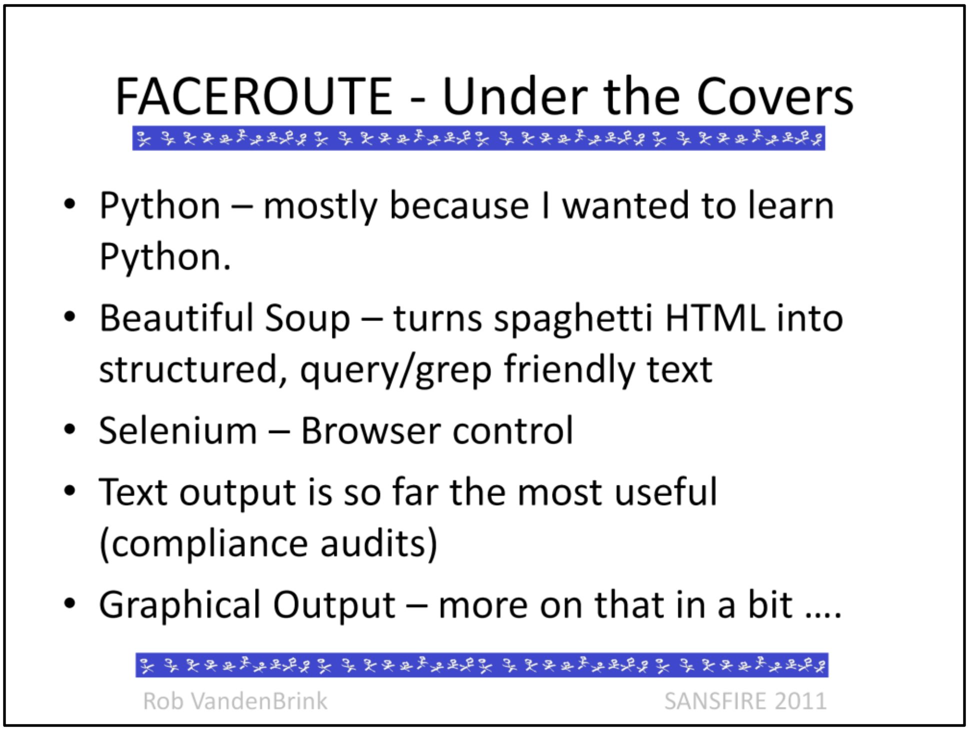
Mechanize is a library that permits stateful browsing. But again, it is foiled by dynamic pages – the page names are not consistently the same, so “finding” the login fields is a problem

So how do we get around the technical problem of dynamic page content?

The answer is simple – if a browser works, to overcome all of this, use a browser!

We’ll drive a standard browser (Firefox) using an automation library. In this version, we used Selenium, powered by Python. Selenium uses a nice client/server interface and so far handles everything needed (except that the PgDn key sequence appears to be broken)

Windmill is an alternate browser that should work for this (I have not tested it).



FACEROUTE - Under the Covers

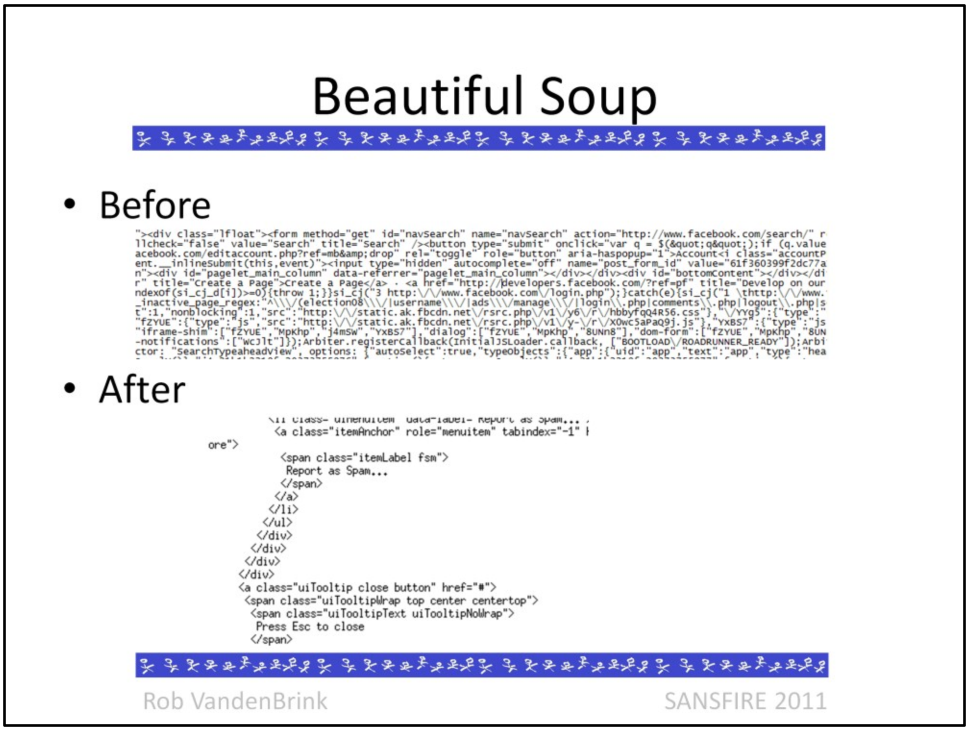
- Python – mostly because I wanted to learn Python.
- Beautiful Soup – turns spaghetti HTML into structured, query/grep friendly text
- Selenium – Browser control
- Text output is so far the most useful (compliance audits)
- Graphical Output – more on that in a bit ….

Rob VandenBrink SANSFIRE 2011

There are a number of moving parts in FACEROUTE

Python – is the overall programming language. PERL would have also worked, but I wanted to learn PYTHON this time around. (please be forgiving on my code).

Beautiful Soup – when scaping the HTML on complex screens, the resulting code is, well, a MESS. Beautiful Soup re-formats the resulting HTML to something we can then use.



Beautiful Soup

- Before

```
"><div class="lfloat"><form method="get" id="navSearch" name="navSearch" action="http://www.facebook.com/search/" r
llcheck="false" value="Search" title="Search" /><button type="submit" onclick="var q = $(&quot;q&quot;);if (q.value
acebook.com/editaccount.php?ref=mb&amp;drop" rel="toggle" role="button" aria-haspopup="1">Account<i class="accountP
ent.__inlineSubmit(this,event)"><input type="hidden" autocomplete="off" name="post_form_id" value="61f360399f2dc77a
n"><div id="pagelet_main_column" data-referrer="pagelet_main_column"></div></div><div id="bottomContent"></div></di
r" title="Create a Page">Create a Page</a> · <a href="http://developers.facebook.com/?ref=pf" title="Develop on our
ndexOf(si_cj_d[i])>=0){throw 1;}}si_cj("3 http:\/\/www.facebook.com\/login.php");}catch(e){si_cj("1 \thttp:\/\/www.
_inactive_page_regex:"^\\\/(election08\\\/|username\\\/|ads\\\/manage\\\/|login\\.php|comments\\.php|logout\\.php|s
t":1,"nonblocking":1,"src":"http:\/\/static.ak.fbcdn.net\/rsrc.php\/v1\/y6\/r\/hbbyfqQ4R56.css"},"\/YYg5":{"type":"
"fZYUE":{"type":"js","src":"http:\/\/static.ak.fbcdn.net\/rsrc.php\/v1\/y-\/r\/XOwc5aPaQ9j.js"},"YxBS7":{"type":"js
"iframe-shim":["fZYUE","MpKhp","j4mSW","YxBS7"],"dialog":["fZYUE","MpKhp","8UNn8"],"dom-form":["fZYUE","MpKhp","8UN
-notifications":["wcJlt"]});Arbiter.registerCallback(InitialJSLoader.callback, ["BOOTLOAD\/ROADRUNNER_READY"]);Arbi
ctor: "SearchTypeaheadView", options: {"autoSelect":true,"typeObjects":{"app":{"uid":"app","text":"app","type":"hea
```

- After

```
              <li class="uiMenuItem" data-label="Report as Spam..." 
               <a class="itemAnchor" role="menuitem" tabindex="-1" h
ore">
                <span class="itemLabel fsm">
                 Report as Spam...
                </span>
               </a>
              </li>
             </ul>
            </div>
           </div>
          </div>
         </div>
         <a class="uiTooltip close button" href="#">
          <span class="uiTooltipWrap top center centertop">
           <span class="uiTooltipText uiTooltipNoWrap">
            Press Esc to close
           </span>
```

Rob VandenBrink SANSFIRE 2011

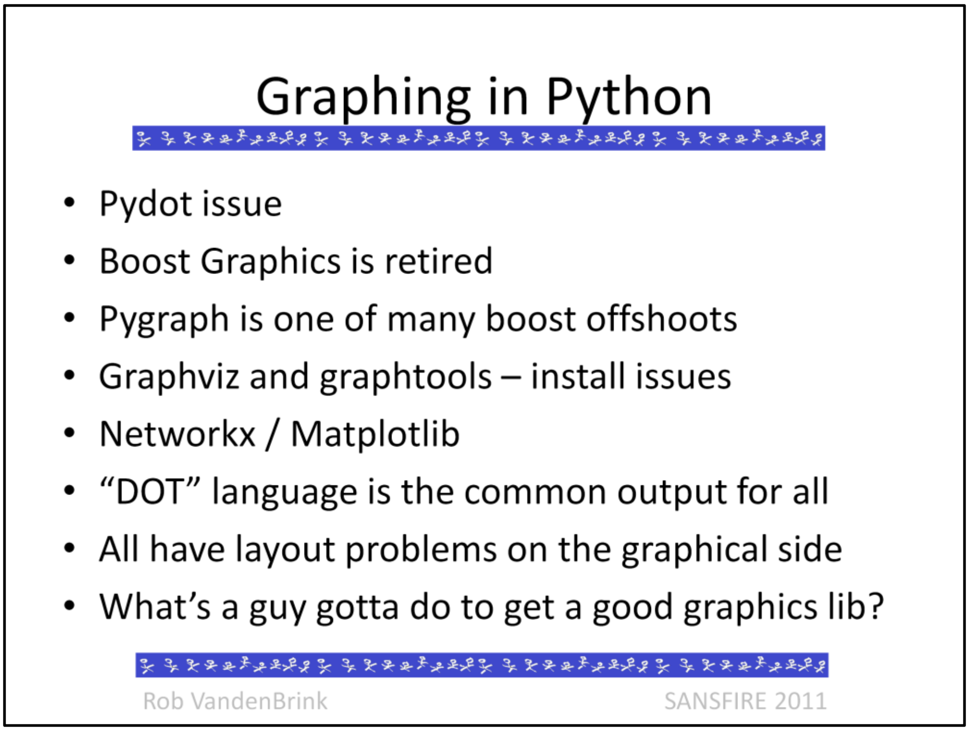
As mentioned, the Beautiful Soup package turns essentially unparseable HTML code into nicely formatted code that can be parsed and manipulated using standard text tools (sort, grep, uniq, etc)

This makes it an ideal tool for scraping and parsing complex HTML screens

For instance, after you've "souped" a friend list page, the command:

'cat out/soup.out | grep "?id=" | grep eng_tid | cut -d\\" -f 2 | grep http'

Will extract the url's of each friend in the list.



Graphing in Python

- Pydot issue
- Boost Graphics is retired
- Pygraph is one of many boost offshoots
- Graphviz and graphtools – install issues
- Networkx / Matplotlib
- “DOT” language is the common output for all
- All have layout problems on the graphical side
- What’s a guy gotta do to get a good graphics lib?

Rob VandenBrink SANSFIRE 2011

Picking a graphics library can be fun in python. Capabilities vary wildly, and some (many? most?) plain just don’t work or won’t install.

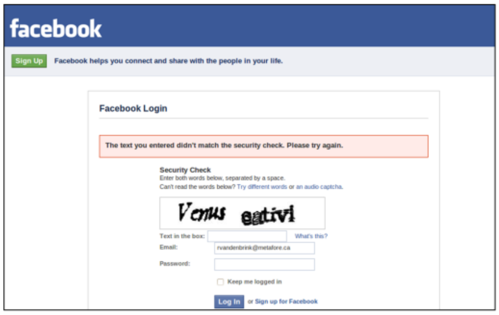
The basic library, pydot, will run out of memory just under 200 nodes. Since it’s common to see facebook users with over 200 friends, this was obviously a no-go.

I ended up using Networkx, which can output XML, DOT (a native graphcis format) or PNG, with matplotlib as a graphics output library

Graphviz has some promise for better output, but installation can be a real challenge !

More Facebook Issues !

- Captchas! Gotta deal with them - ‘ nuff said …



facebook

Sign Up Facebook helps you connect and share with the people in your life.

Facebook Login

The text you entered didn't match the security check. Please try again.

Security Check
Enter both words below, separated by a space.
Can't read the words below? Try different words or an audio captcha.

Text in the box: What's this?
Email: rvandenbrink@metafore.ca
Password:

□ Keep me logged in

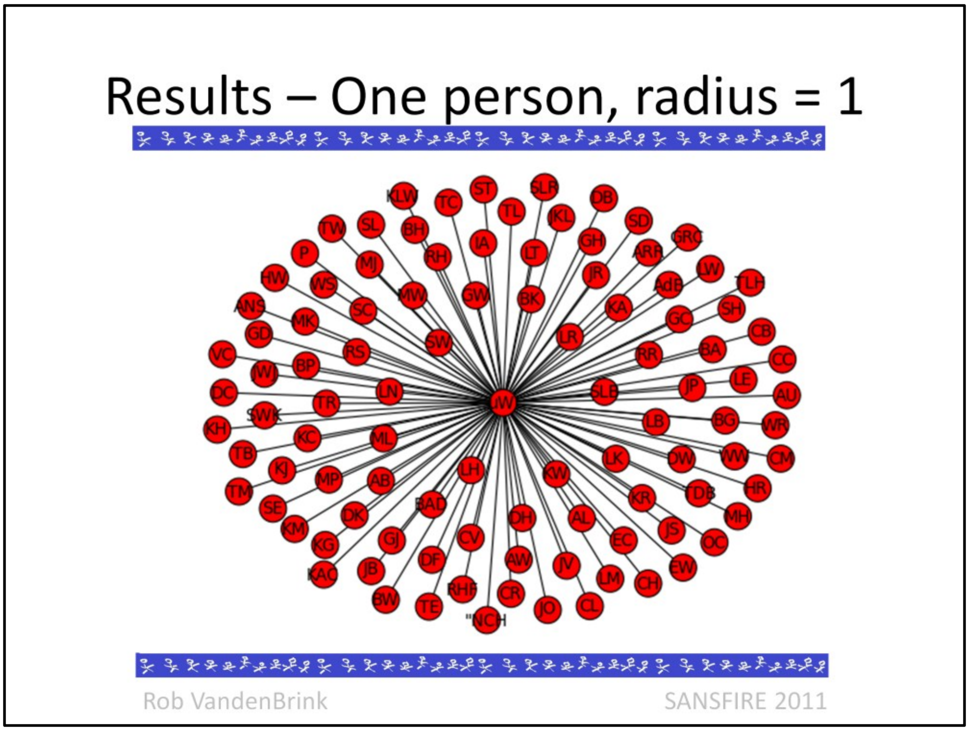
Log In or Sign up for Facebook

Those people at Facebook aren’t dumb !

As soon as you trigger a number of queries (it varies), you start getting captcha screens.

How do we deal with it in the code? Monitor for the condition, alert and wait for manual intervention that’s how !

```
while 'Security Check' in  repr(sel.get_html_source()):
        print "captcha"
        time.sleep(15)
```



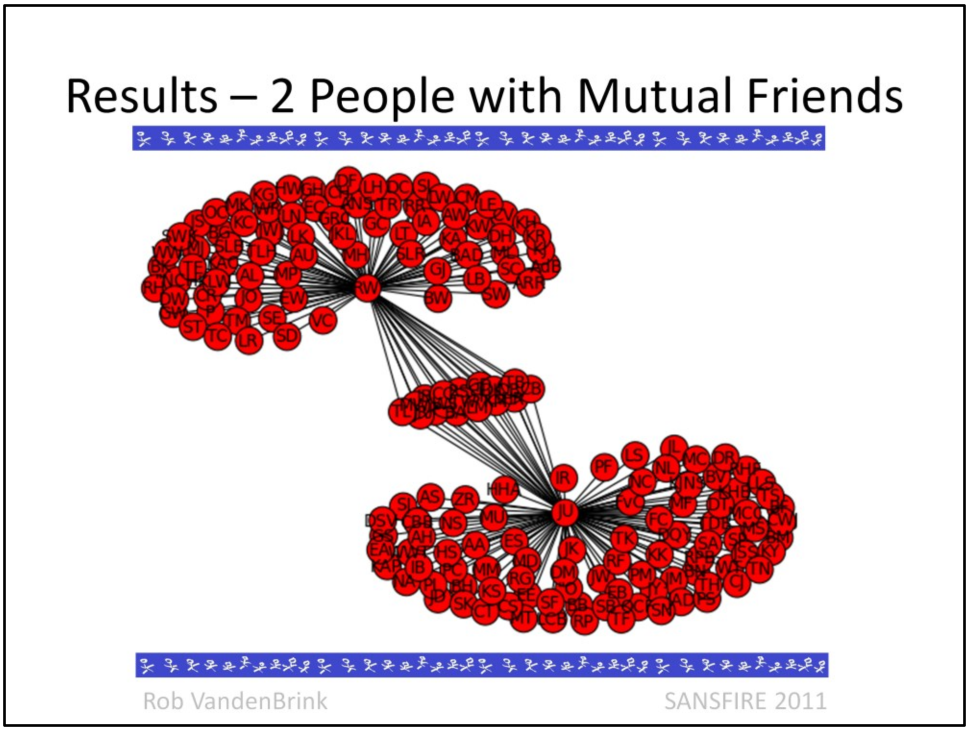
Results – One person, radius = 1

Rob VandenBrink

SANSFIRE 2011

This shows a single facebook user with several friends (radius = 1)

Initials are used for anonymity, as well we clarity of the resulting graphic

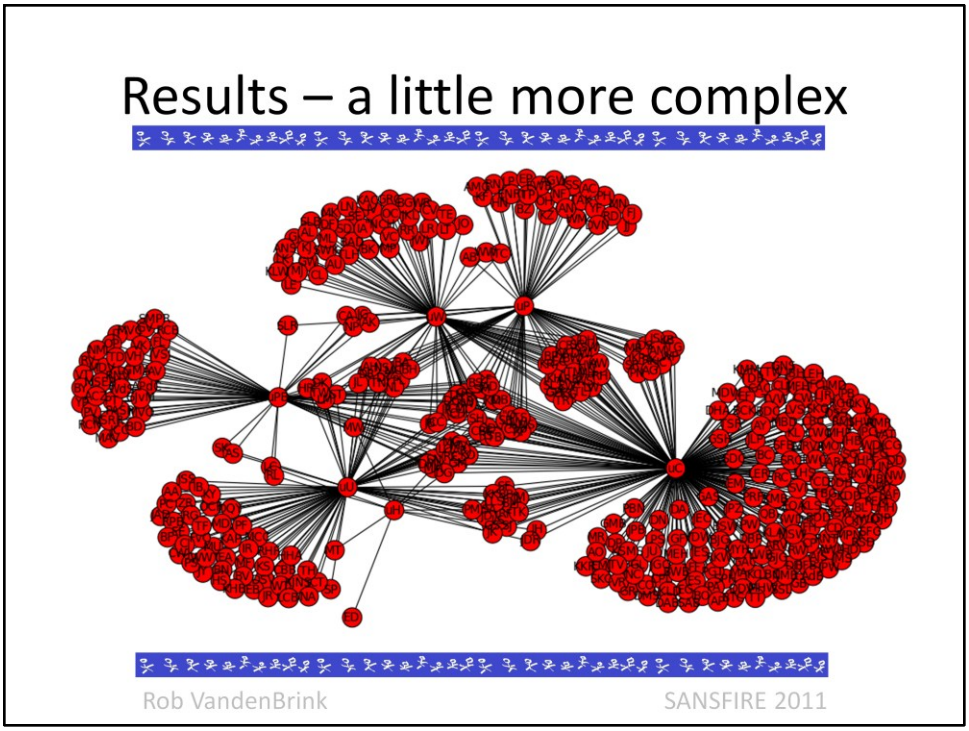


Results – 2 People with Mutual Friends

Rob VandenBrink

SANSFIRE 2011

This graph shows two people with a large number of mutual friends

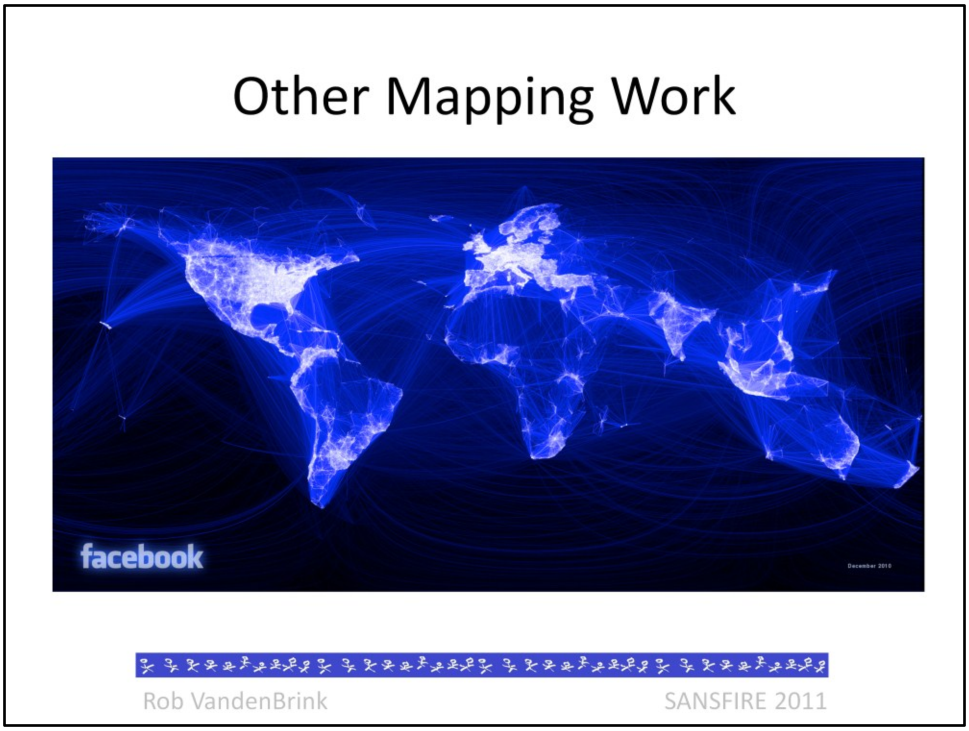


Results – a little more complex

Rob VandenBrink

SANSFIRE 2011

You can see that the graphical output gets complex very quickly.
What I'd like to see in a final selection:
More control over colour and weights – as density increases the graph quickly becomes a black "blob". The final graph should have target nodes highlighted, and everything else in a much lighter colour (20% grey or there-abouts) to show pattern but not obscure the targets.
More control over node shape – it would be nice for instance to implement a "stop sign" shape for nodes that have their security settings tweaked to not display friends.
More control over the size of the canvas. The current limits mean that we're using initials just to fit the graphs on the screen.

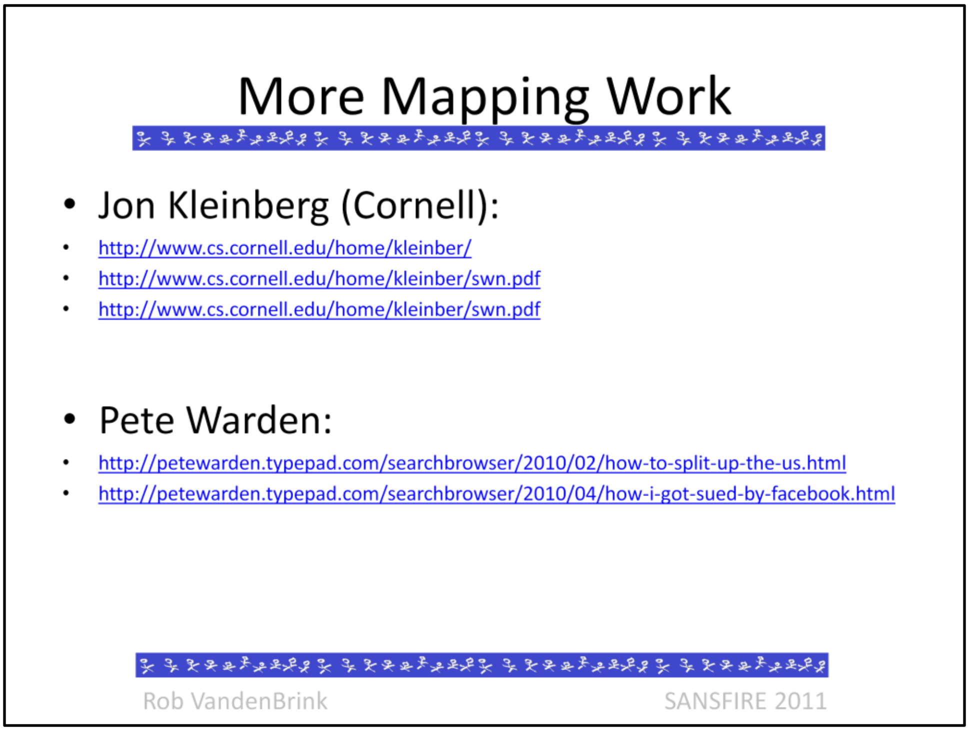


Other Mapping Work

Rob VandenBrink SANSFIRE 2011

Facebook Intern Paul Butler’s created a “Facebook Map of the World”, which maps out friendships across the world using the back-end datastore at Facebook.
Of course, one of the challenges was “too much information” – just mapping all links resulted in a white blob.
Used a concept of statistical “city link weight” to reduce inter-city lines to the point where the map was readable.
Longer links are displayed as great circle arcs for aesthetic (and accuracy) reasons

More info on Paul’s work can be found here:
http://www.facebook.com/note.php?note_id=469716398919



More Mapping Work

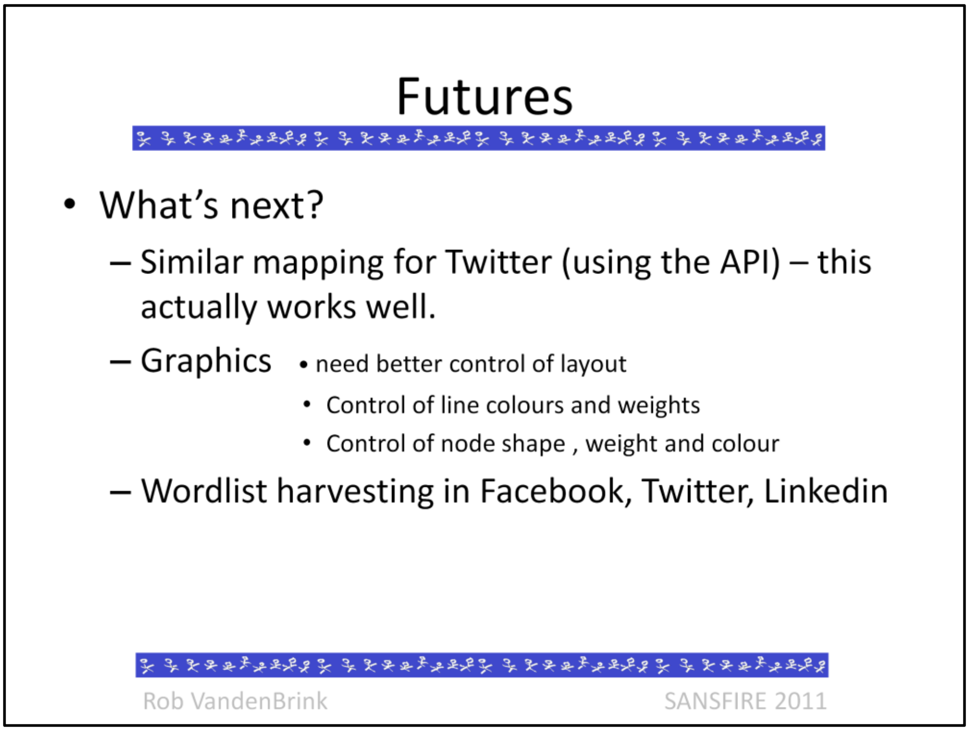
- Jon Kleinberg (Cornell):
  - http://www.cs.cornell.edu/home/kleinber/
  - http://www.cs.cornell.edu/home/kleinber/swn.pdf
  - http://www.cs.cornell.edu/home/kleinber/swn.pdf

- Pete Warden:
  - http://petewarden.typepad.com/searchbrowser/2010/02/how-to-split-up-the-us.html
  - http://petewarden.typepad.com/searchbrowser/2010/04/how-i-got-sued-by-facebook.html

Rob VandenBrink SANSFIRE 2011

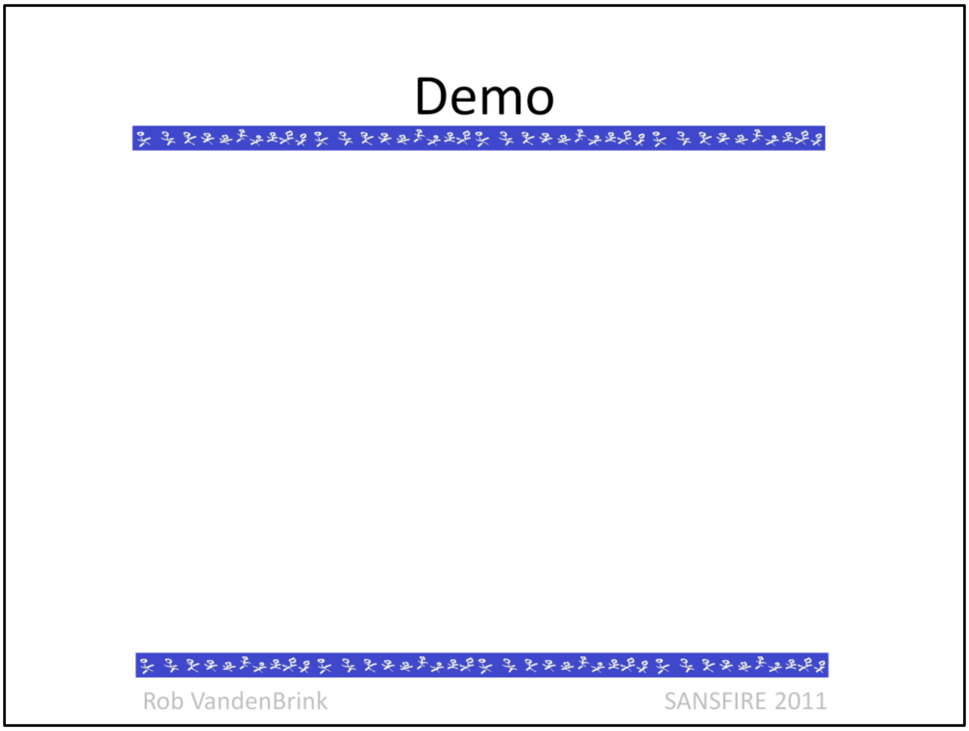
Jon Kleinburg of Cornell remains one of the primary researchers into mapping and interpreting information from complex networks into useful formats. His work provided the inspiration for SMRT – while SMRT is primarily a data collection and representation toolset, Jon's work presumes that you have this, and starts from there.

Pete Warden did some pioneering work in this area of data collection and representation as well, and got sued by Facebook for his trouble. **Crawling and screen-scraping Facebook does violate their terms of service, please do keep this in mind while weighing the practical use of tools like FACEROUTE**



Futures

- What's next?
  - Similar mapping for Twitter (using the API) – this actually works well.
  - Graphics
    - need better control of layout
    - Control of line colours and weights
    - Control of node shape , weight and colour
  - Wordlist harvesting in Facebook, Twitter, Linkedin



Demo