Hacking Piazza

or: How I Learned to Stop Worrying and Love TOS (violations)

Mike Clark | University of Calgary

Piazza


piazza
Log in
The (Free) Efficient Way to Manage Class Q&A
How is this better than email, newsgroups, and discussion forums?
Students actually use Piazza, they love it. This difference stems from how we built Piazza. We've personally met with and spoken to thousands of students and instructors. The result is a beautifully intuitive and simple product that students love and use.
question
Finding normal vector for surface
the students' answer,
the instructors' answer,
Student?
Search your classes
Instructor?
Create or Join your class
View a Demo Class
Interested in using Piazza for groups other than a class?
Learn More
Classes being created on Piazza:
ME- 4340-A ME-4340-A
Minami Yoda | 20 students
15-112 Fundamentals of Programming and Computer Science, Fall 20…
Michaela Van Peursem, AJ Kaufmann | 200 students
CS 9B (SELF-PACED) Self-paced Pascal for Programmers
Dan Garcia, Carol | 25 students
ECE 190 Intro to Computing Systems
Grace Goo, Venkatasai Koppula | 300 students
Read about Piazza in... TechCrunch The New York Times MercuryNews (See All)

Tools

- Microsoft Windows 7 x64 SP1 (v. 6.1.7601)
- Oracle VM Virtual Box Manager (v.4.2.10 r84104)
  - A virtualization software package developed by Sun Microsystems.
- Kali Linux (kali-linux-1.0.1-amd64)
  - A distribution based on the Debian GNU/ Linux distribution aimed at digital forensics and penetration testing use.
- Notepad++ (v.5.9.8-UNICODE)
  - A free source code editor which supports several programming languages
- puTTY (v.0.62)
  - A free implementation of Telnet and SSH for Windows and Unix platforms, along with an xterm terminal emulator.
- Google Chrome Web Browser (v. 26.0.1410.43 m)
  - Chrome Developers Tools
    - Provide web developers deep access into the internals of the browser and their web application.
- Mozilla Firefox Web Browser (19.02-R)
  - Firebug (v.1.11.2)
    - Developer toolkit for Mozilla Firefox.
  - Tamper Data (v.11.0.1)
    - A tool to view and modify HTTP/HTTPS headers and post parameters.
  - HttpRequester (v.1.0.4)
    - A tool for easily making HTTP requests, viewing the responses, replaying and keeping a history of transactions.
- Misc. Homebrew Scripts & Programs

Web Application Review

- Client Side
  - Renders HTML documents served by Piazza
  - Interprets CSS for styling
  - Executes JavaScript
    - dynamic content generation
    - Client-side validation
    - Primary functionality encapsulated within *dashboardXXXX.js*
  - Dispatches XMLHttpRequests (XHR) for asynchronous communication
- Server Side
  - Deliver CSS
  - Deliver JavaScript
  - Generates and serves HTML documents
  - Dynamically generated from:
    - Internal state of the system
    - User Input
      - Communicated via XHR

Reconnaissance

- Web Server
  - Recon info available in response headers and network traffic
  - This may be just one of many, or a load balancer
  - Northeastern U.S.A.
  - Amazon EC2
    - IP: 54.243.185.105 (at the time)
  - nginx/1.2.1.16
    - HTTP and reverse proxy server, as well as a mail proxy server.
    - Used by 12.96% busiest sites in April 2013[1]
  - Phusion Passenger (mod_rails/ mod_rack) 3.0.7
- API Server
  - Recon info available in response headers and network traffic
  - Handles API requests
  - Northeastern U.S.A.
  - Amazon EC2
    - IP: 54.243.185.105 (at the time)
    - CloudFront sitting on top
  - nginx/1.2.1.1345
  - Jetty/v.?.?.?
    - Web server and javax.servlet container
    - support for SPDY, Web Sockets, OSGi, JMX, JNDI, JASPI, AJP and many other integrations.

[1] http://goo.gl/qBJsD

Reconnaissance

Background

- Initially interested in performing unauthorized actions
  - Double-voting, impersonating other users
- Looked at circumventing client-side validation
  - Trivial!
  - Digging through client-side JavaScript and analyzing network traffic allowed me to analyze and enumerate the underlying REST API
- Why not just send forged requests directly to the API?

REST API

- Allow the user to communicate with the system by way of HTTP GET/POST requests
- Bi-Directional Communication
  - Request and Post parameters: Pass data in
  - Response body: Pass data out
- Traditionally, requests and responses are engineered around the transfer of resource representations to allow the client to transition between states
- However, with XHR requests occurring in the background, the system is in a constant state of flux

REST API

- URL Format
  http://piazza.com/logic/api?method=[TYPE]&aid=[HASH]
  - Content
    - Content.create , Content.get, Content.vote , Content.answer …
  - Network
    - Network.create , Network.get_all_users , Network.update …
  - User
    - User.get_user_profile , User.update_user_profile , User.update …

REST API

- URL Format
  http://piazza.com/logic/api?method=[TYPE]&aid=[HASH]
  - Generated as follows (JavaScript):
    (new Date()).getTime().toString(36) \
    +Math.round(Math.random()*1679616).toString(36);
  - Base 36 Encoding [a-z,0-9]
  - Concatenation of:
    - Current UNIX Epoch timestamp (milliseconds elapsed since 1 Jan, 1970 UTC)
    - Nonce of range: [0-1679616)

- POST data (JSON)
  {“method”:“[type]”,”params”:{“param1”:”value1”,…}}

REST API

- POST Data
  - Contained in the HTTP body
  - JavaScript Object Notation (JSON) encoding
  - Format {“method”:“[type]”,”params”:{“param1”:”value1”, “param2”:”value2”,…}}
  - Examples:
    - {"method":"content.get","params":{"cid":"hf93123l4be6ei","nid":"hf4zsf2w7jv5ri"}}
    - {"method":"network.get_users","params":{"ids": ["gxm5422kbqz4yi"],"nid":"hf4zsf5ri”}}
    - {"method":"user.get_user_profile","params": {"nid":"hf41234w7jv5ri","uid":"hf425we5d6r2fu","preview_profile":false,"campaign_id":"click-selfProfileFromTopbar"}}:

REST API

- Response Data
  - Contained in the HTTP response body
  - JavaScript Object Notation (JSON) encoding
  - Format {“method”:“[type]”,”params”:{“param1”:”value1”, “param2”:”value2”, …}}
  - Examples:
    - **Content.edit**
      {"result":"hfjqv458fxb4k9","error":null,"aid":"hfjqvok0eol3"}
    - **Network.get_users**
      {"result":[{"id":"hf5c11237zv2qd","admin":false,"name":"Dumme Dave Aunt","role":"student","us":false,"facebook_id":null,"photo":null},{"id":"hf5568w8zzp32m","admin":false,"name":"Dumme Dave Sister","role":"","us":false,"facebook_id":“123","photo":null}],"error":null,"aid":"hfjquoy427db"}
    - **User.update_user_profile**
      {"result":"OK","error":null,"aid":"hfjqvu7rmf1c"}

REST API

- Four primary pieces of meta-data used to communicate with the API
  - Nonce? *(aid)*
    - Sent with ever request
    - Generated as discussed earlier
    - Unsure of actual use; re-using/random values does not seem to have an affect
  - User ID *(uid)*
    - Used to identify the user
    - Generated similar to aid
    - Instead of random value appended to the date, it is sequential number (proven through experimentation)
  - Network ID *(nid)*
    - Used to identify a network (course)
    - Generated similar to aid
    - Could not determine if the second half of the hash is random or sequential
  - Content ID *(cid)*
    - Sent with ever request
    - Generated similar to aid

Reconnaissance

piazza | CPSC 525.14 10 | Q & A | Course Page | Dumme Daves Uncle

hw1 hw2 hw3 hw4

Unread Updated Unresolved Following

+ New Post | Search or add a post...

TODAY

vhg S

LAST WEEK

Post 2 1

Post 1 1

change picture

Dumme Daves Uncle

University of Calgary

+ Add one line that best describes you

1 views

Preview Profile

Academics

Degree + Add

Grad Year + Add

Major/Dept. + Add

Courses

Winter 2013 CPSC 525.14

525.13

+ Add a class

Elements Resources Network Sources Timeline Profiles Audits Console

Name / Path

api?method=content.create&aid=hfjqvitionge /logic

api?method=content.get&aid=hfjqvj8u1a8k /logic

api?method=content.edit&aid=hfjqvok0eol3 /logic

api?method=content.answer&aid=hfjqvtyu9saq /logic

api?method=content.view&aid=hfjqvu7rmf1c /logic

api?method=user.get_user_profile&aid=hfjqwg4z8k9i /logic

api?method=user.update_user_profile&aid=hfjqx4d71exz /logic

api?method=network.get_online_users&aid=hfjqxzcgx64k /logic

api?method=user.update_user_profile&aid=hfjqycp5etvw /logic

api?method=network.get_online_users&aid=hfjr1um72dgr /logic

api?method=network.get_online_users&aid=hfjr5pv4hsxj /logic

api?method=network.get_online_users&aid=hfjr9l2njk2s /logic

api?method=network.get_online_users&aid=hfjrdg9zj7tu /logic

api?method=network.get_online_users&aid=hfjrhbfogtvg /logic

api?method=network.get_online_users&aid=hfjrl6ng337z /logic

api?method=network.get_online_users&aid=hfjrp1wooeqv

Headers Preview Response Cookies Timing

Request URL: https://piazza.com/logic/api?method=content.get&aid=hfjqvj8u1a8k
Request Method: POST
Status Code: 200 OK

Request Headers view source

Accept: application/json, text/javascript, */*; q=0.01
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-GB,en-US;q=0.8,en;q=0.6
Connection: keep-alive
Content-Length: 81
Content-Type: application/x-www-form-urlencoded
Cookie: [redacted]
Host: piazza.com
Origin: https://piazza.com
Referer: https://piazza.com/class
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31
X-Requested-With: XMLHttpRequest

Query String Parameters view source view URL encoded

method: content.get
aid: hfjqvj8u1a8k

Form Data view source view URL encoded

{"method":"content.get","params":{"cid":"hfjqvjlcqep4hx","nid":"hf4zsf2w7jv5ri"}}

Response Headers view source

Connection: close
Content-Encoding: gzip
Content-Type: text/html;charset=UTF-8
Date: Mon, 15 Apr 2013 14:35:24 GMT
Expires: Thu, 01 Jan 1970 00:00:00 GMT
P3P: CP='IDC DSP CURa ADMa DEVa OUR ONL UNI COM NAV'
Server: nginx/1.2.1.1431
Set-Cookie: [redacted]
Transfer-Encoding: chunked
Vary: Accept-Encoding

Reconnaissance

Now that we have discussed how the API is implemented

How can we abuse it?

Session Keys

Session Keys

- Piazza determines authorization primarily through the use of Session Keys

Session Keys

- Piazza determines authorization primarily through the use of Session Keys
  - Recall: HTTP is a state-less protocol
    - Each authenticated user has a *session key* associated with them
    - Server keeps a copy; Client keeps another copy
    - Communicated via a *cookie* over top of an established *https* connection
    - Whenever authorization to perform an action (eg. view a network, comment on a post) is checked, the user is authenticated using the session key
    - A Session key is sent with each HTTP request

Session Keys

- Why not just send the *uid* of the logged in user with each request?

Session Keys

- Why not just send the *uid* of the logged in user with each request?

Simple

Never Trust the User.

If a malicious user wanted to perform an action they were not authorized to, they could simply substitute the *uid* of a user known to have that level of authorization.

Session Keys

Vulnerability?

- Session Keys over a week old appear to be still valid
  - After logging out
  - After closing the web browser
  - After client-machine restarts
- Hypothesis: Session keys are not being destroyed properly server-side
- Conclusion: Something is amiss…. Not enough time to fully investigate.

Session Keys

Piazza – Ask. Answer. Explore. Whenever. - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Online Scheduler | Piazza – Ask. Answer. Explore....

https://piazza.com

piazza

Log in

Email

mbclark@ucalgary.ca

Password

••••••••

Keep me logged in

Forgot your password?

Log in Cancel

The (Free) ... s Q&A

How is this better t...

Students actually use ... Piazza. We've personal... instructors. The result ... and use.

Student? Instructor?

Search your classes

Create or Join your class

Console HTML CSS Script DOM Net Cookies

Clear Persist Profile All Errors Warnings Info Debug Info Cookies

```
>>> PA
Object { users={...}, userQueue=[0], ajaxLogin=false, more... }
>>> PA.user
null
>>>
```

HttpRequester

Request

URL: https://piazza.com/logic/api?aid=hf6cgzj5ft6m

POST Submit GET POST PUT New request Paste Request

Content to Send Headers Parameters

Content Type: application/x-www-form-urlencoded

Content Options: Base64 Parameter Body

Content File Browse...

{"method":"network.get_users","params":{"ids":["hf5cy3lp15o3ad"],"nid":"hf4zsf2w7jv5ri"}}

Response

POST on https://piazza.com/logic/api?aid=hf6cgzj5ft6m

Status: 200 OK View raw transaction

{"result":[{"id":"hf5cy3lp15o3ad","admin":false,"name":"Dummer Daves Brother","role":"student","us":false,"facebook_id":"730431411","photo":null}],"error":null,"aid":"hf6cgzj5ft6m"}

Headers:

Server	nginx/1.2.1.16
Date	Sat, 06 Apr 2013 06:57:35 GMT
Content-Type	text/html;charset=UTF-8
Transfer-Encoding	chunked
Connection	close
Vary	Accept-Encoding
Set-Cookie	piazza_session="GwwtlxlLtwHxCL
Expires	Thu, 01 Jan 1970 00:00:00 GMT
p3p	CP="IDC DSP CURa ADMa DEVa

History

Request	Response	Date	ContentLength	ElapsedTime
POST https://piazza.com/logic/api?aid=hf6cgzj5ft6m	200 OK	Apr 6 2013 - 12:57:29 AM	180	433 ms
POST https://piazza.com/logic/api?aid=hf6cgzj5ft6m	200 OK	Apr 6 2013 - 12:51:43 AM	86	338 ms
POST https://piazza.com/logic/api?aid=hf6cgzj5ft6m	200 OK	Apr 6 2013 - 12:50:53 AM	86	317 ms
POST https://piazza.com/logic/api?aid=hf6cgzj5ft6m	200 OK	Apr 6 2013 - 12:48:28 AM	86	434 ms
POST https://piazza.com/logic/api?aid=hf6cgzj5ft6m	200 OK	Apr 6 2013 - 12:48:10 AM	86	400 ms
POST https://piazza.com/logic/api?aid=hf6cgzj5ft6m	200 OK	Apr 5 2013 - 11:40:20 PM	86	431 ms
POST https://piazza.com/logic/api?aid=hf54fz2k13200	200 OK	Apr 5 2013 - 11:29:02 PM	50	362 ms
POST https://piazza.com/logic/api?aid=hf54fz2k13200	200 OK	Apr 5 2013 - 3:01:58 AM	50	392 ms

Clear history | Copy to clipboard | Delete request | Edit raw request... | Save request... | Load request...

KNOW

REVOLT

ENSLAVEMENT

you are headed here

1:01 AM

Session Keys

What else can we do?

Smurf Accounts

Smurf Accounts

- Gmail ignores periods in the username

Smurf Accounts

- Gmail ignores periods in the username
  - ace@gmail.com
  - a.ce@gmail.com
  - ac.e@gmail.com
  - a.c.e@gmail.com
  - a.c.e.@gmail.com
  - a.c.e..@gmail.com
  - a.c.e...@gmail.com
  - ace...@gmail.com
  - a.ce...@gmail.com
  - ac.e...@gmail.com
  - a.c.e...@gmail.com
  - a.c.e....@gmail.com
  - a.c.e.....@gmail.com
  - a.c.e.....@gmail.com

Smurf Accounts

- Gmail ignores periods in the username
  - ace@gmail.com
  - a.ce@gmail.com
  - ac.e@gmail.com
  - a.c.e@gmail.com
  - a.c.e.@gmail.com
  - a.c.e..@gmail.com
  - a.c.e...@gmail.com
  - ace...@gmail.com
  - a.ce...@gmail.com
  - ac.e...@gmail.com
  - a.c.e...@gmail.com
  - a.c.e....@gmail.com
  - a.c.e.....@gmail.com
  - a.c.e.....@gmail.com

Piazza does not!

Smurf Accounts

- What does this mean?
  - We can create a tonne of Piazza accounts with only one (maybe legit?) email account!

Smurf Accounts

- What does this mean?
  - We can create a tonne of Piazza accounts with only one (maybe legit?) email account!
- But won't I still have to create each Piazza account individually?
  - Luckily for us, Piazza makes it easy to create multiple accounts; provided we have unique emails to feed it (or at least, what Piazza *thinks* are unique emails)

Smurf Accounts

General Settings | Customize Q&A | **Manage Enrollment** | Create Groups | Customize Course Page

Have all of your students been enrolled yet? Has each activated his/her account? You can always re-send the welcome emails to students who have not yet activated their Piazza accounts.

Enroll Students

Copy and paste email addresses in any format.
↳ Once added, they receive a Welcome email with a link to activate their Piazza account.

☐ Enable Add/Drop

Add Students or upload a file

Student Roster: ...out of 2052 (estimated) Edit

2052 enrolled

The following students are enrolled in your class.
2040 students have not activated their accounts. Resend activation email?
↳ Those who've activated their accounts will have names alongside their email addresses.

- ☐ t.hi.si.sa.d.umm.e@gmail.com
- ☐ th.i.s.is.adu.m.m.e@gmail.com
- ☐ t.his.is.a.d.um.m.e@gmail.com
- ☐ thi.si.s.ad.u.mme@gmail.com
- ☐ th.is.isa.du.mm.e@gmail.com
- ☐ th.i.s.i.s.adum.me@gmail.com
- ☐ thi.s.is.adu.m.m.e@gmail.com
- ☐ t.hi.s.is.a.d.um.me@gmail.com
- ☐ t.hi.s.i.s.a.du.m.m.e@gmail.com

Smurf Accounts

General Settings | Customize Q&A | **Manage Enrollment** | Create Groups | Customize Course Page

Have all of your students been enrolled yet? Has each activated his/her account? You can always re-send the welcome emails to students who have not yet activated their Piazza accounts.

Enroll Students

Copy and paste email addresses in any format.
↳ Once added, they receive a Welcome email with a link to activate their Piazza account.

☐ Enable Add/Drop

Add Students or upload a file

Student Roster: ...out of 2052 (estimated) Edit

2052 enrolled

The following students are enrolled in your class.
2040 students have not activated their accounts. Resend activation email?
↳ Those who've activated their accounts will have names alongside their email addresses.

- ☐ t.hi.si.sa.d.umm.e@gmail.com
- ☐ th.i.s.is.adu.m.m.e@gmail.com
- ☐ t.his.is.a.d.um.m.e@gmail.com
- ☐ thi.si.s.ad.u.mme@gmail.com
- ☐ th.is.isa.du.mm.e@gmail.com
- ☐ th.i.s.i.s.adum.me@gmail.com
- ☐ thi.s.is.adu.m.m.e@gmail.com
- ☐ t.hi.s.is.a.d.um.me@gmail.com
- ☐ t.hi.s.i.s.a.du.m.m.e@gmail.com

Smurf Accounts

General Settings | Customize Q&A | **Manage Enrollment** | Create Groups | Customize Course Page

Have all of your students been enrolled yet? Has each activated his/her account? You can always re-send the welcome emails to students who have not yet activated their Piazza accounts.

Enroll Students

Copy and paste email addresses in any format.
↳ Once added, they receive a Welcome email with a link to activate their Piazza account.

☐ Enable Add/Drop

Add Students or upload a file

Student Roster: ...out of 2052 (estimated) Edit

2052 enrolled

The following students are enrolled in your class.
2040 students have not activated their accounts. Resend activation email?
↳ Those who've activated their accounts will have names alongside their email addresses.

- ☐ t.hi.si.sa.d.umm.e@gmail.com
- ☐ th.i.s.is.adu.m.m.e@gmail.com
- ☐ t.his.is.a.d.um.m.e@gmail.com
- ☐ thi.si.s.ad.u.mme@gmail.com
- ☐ th.is.isa.du.mm.e@gmail.com
- ☐ th.i.s.i.s.adum.me@gmail.com
- ☐ thi.s.is.adu.m.m.e@gmail.com
- ☐ t.hi.s.is.a.d.um.me@gmail.com
- ☐ t.hi.s.i.s.a.du.m.m.e@gmail.com

Smurf Accounts

- Now that we have all these Smurf Accounts, what can we do with them?

Smurf Accounts

- Now that we have all these Smurf Accounts, what can we do with them?

Some very interesting things….

Merging users

Merging Users

- Piazza allows for users to merge two accounts

Merging Users

- Piazza allows for users to merge two accounts
  - Can be done during account registration OR from an already-existing account
  - All that is required is the email address of the account to be merged with
  - The account that the merge is initiated from becomes the *dominate* account

Domination

Suppose user **ALICE** initiates a merge with **BOB**

Merging Users

Domination

Suppose user **ALICE** initiates a merge with **BOB**

- **ALICE** inherits almost all of **BOB**s attributes
  - Profile Picture
  - Networks
    - **BOB** is effectively kicked out of his networks
  - Admin privileges in networks
  - Content submission list
  - A lot of other attributes used by Piazza behind the scenes

Merging Users

Domination

Suppose user **ALICE** initiates a merge with **BOB**

- **ALICE** inherits almost all of **BOB**s attributes
  - Profile Picture
  - Networks
    - **BOB** is effectively kicked out of his networks
  - Admin privileges in networks
  - Content submission list
  - A lot of other attributes used by Piazza behind the scenes
- What **ALICE** does not inherit from **BOB**
  - User ID (*uid*)
  - *Actual* content postings
  - Statistics
  - Login Credentials
    - **BOB** will inherit **ALICE**'s
  - Facebook Attributes

Merging Users

Domination

Suppose user **ALICE** initiates a merge with **BOB**

- End Result?

Merging Users

Domination

Suppose user **ALICE** initiates a merge with **BOB**

- End Result? **BOB** is now essentially a zombie account.
  - Account and *uid* is retained on the system BUT…

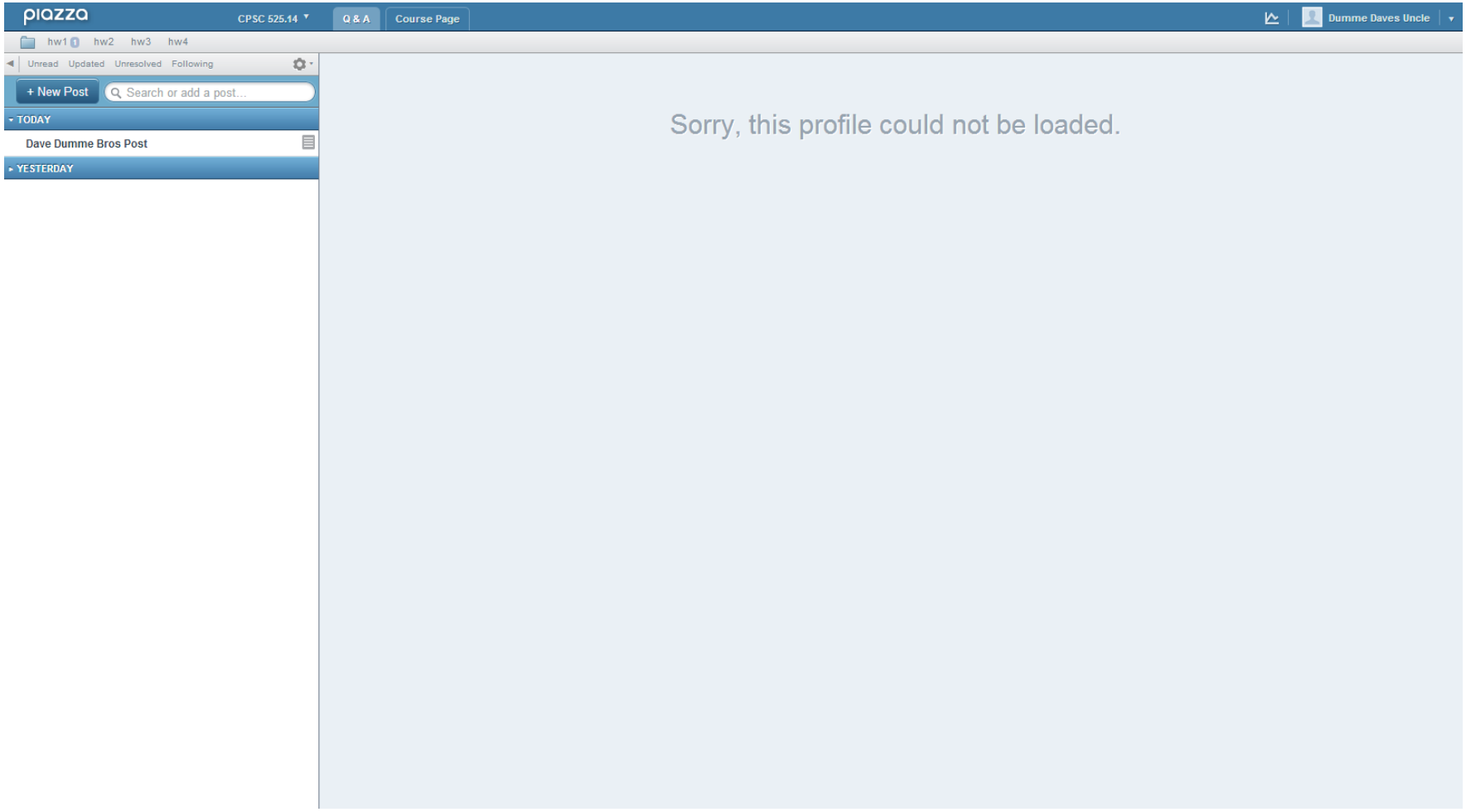
Merging Users

Domination

Suppose user **ALICE** initiates a merge with **BOB**

- End Result? **BOB** is now essentially a zombie account.
  - Account and *uid* is retained on the system BUT…
  - User is essentially locked out
    - Attempts to login with either **ALICE**'s or **BOB**'s credentials will sign into **ALICE**'s account.
    - Could not find a way to undo merges
    - But we can still drop classes, effectively removing access to that network for both **ALICE** and **BOB** (denial of service)

Merging Users



Merging Users

- Great, I can lock myself out of my own account just like *Dave the Dummy*...

So What?

Merging Users

- Gaining entrance into networks without the admin's consent

Merging Users

- Gaining entrance into networks without the admin's consent
  - Recall: The merged account inherits all the networks
  - Thus, you can get users with new *uid*'s into a network you were already in

Merging Users

- Gaining entrance into networks without the admin's consent
  - Recall: The merged account inherits all the networks
  - Thus, you can get users with new *uid*'s into a network you were already in
- Not only that, but because the merged user has a new *uid*, so they can now replay all the activities of the first user (eg. voting again in polls)

Merging Users

- BUT WAIT…. THERE'S MORE!!!

Merging Users

- When merging, we neglected the details about how the merge is confirmed by the other party.

  Recall: **ALICE** initiates a merge with **BOB** by providing his email address associated with his Piazza account.

Merging Users

piazza | CPSC 525.14 10 | Q & A | Course Page | Dumme Daves Uncle

Account Settings

Personal Settings

Full Name: Dumme Daves Uncle

Password: Change password

Preferred Email: t.h.is.isadu.m.m.e@gmail.com

Other Emails: thi.sisad.um.me@gmail.com

this.is.a.d.u.m.me@gmail.com

Save | Cancel

Save Profile

change picture

Log In

OR use Facebook photo + stay connected (we'll never post to your profile)

Add Email Accounts

We'll send a confirmation link to the email address you entered. Any classes associated with that address will be added to this account.

Send Link | Cancel

Class & Email Settings

525.13 | 525.BUG_CHECK | WINTER 2013

Smart Digest | Real Time | Edit Email Notifications | Drop Class

CPSC 525.14 | CPSC 525.42: Bug 2 | WINTER 2013

Smart Digest | Real Time | Edit Email Notifications | Drop Class

Universal Access Preferences

I want to use an assistive device for visual or motor impairments with Piazza

Save Accessibility Preferences

Merging Users

- When merging, we neglected the details about how the merge is confirmed by the other party.

  Recall: **ALICE** initiates a merge with **BOB** by providing his email address associated with his Piazza account.

  - **BOB** receives an email informing him of a merge request

Merging Users

Piazza Team 5:25 AM (0 minutes ago)

to this.is.a.d.u..

Hello!

We received a request to merge this.is.a.d.u.m.me@gmail.com to your t.h.is.isadu.m.m.e@gmail.com account on Piazza.

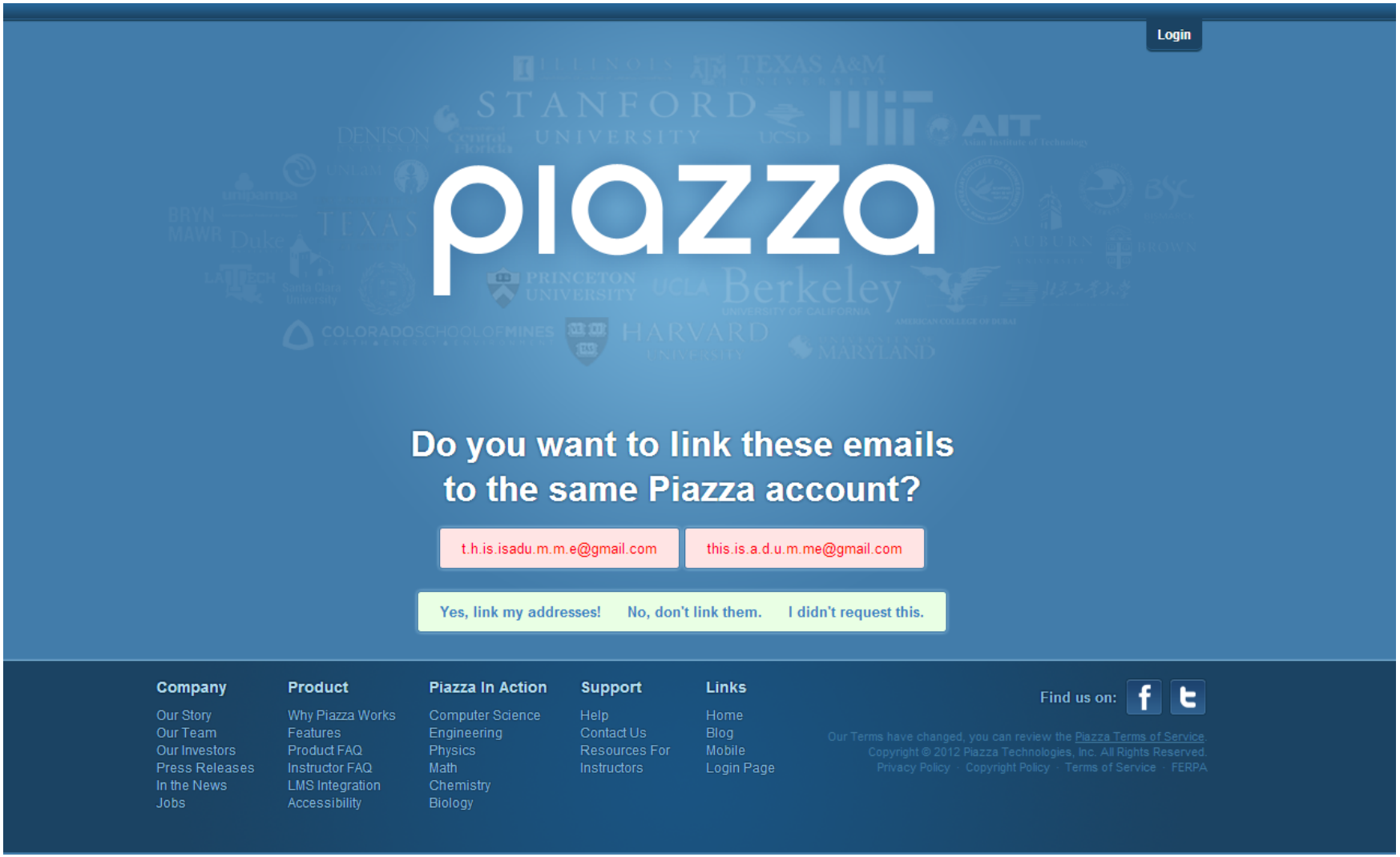
To confirm this request, follow this link: https://piazza.com/verify_email/hf5cxwe5d6r2fu/hfi4mzpmmhy51p?preferred=no

...

Click here to Reply, Reply to all, or Forward

- A link is provided that will bring **BOB** to a web page with the option to
  - Accept
  - Decline
  - Report to Piazza this was a *bad* request
- No, authentication required. Just knowledge of the URL

Merging Users



Merging Users

- How is this merge request link generated?

https://piazza.com/verify_email/hf5cxrptw2a1vm/hf6rag7dikg5lb?preferred=yes

Merging Users

- How is this merge request link generated?

https://piazza.com/verify_email/hf5cxrptw2a1vm/hf6rag7dikg5lb?preferred=yes

ALICEs *uid*

Merging Users

- How is this merge request link generated?

https://piazza.com/verify_email/hf5cxrptw2a1vm/hf6rag7dikg5lb?preferred=yes

ALICEs *uid*

HASH code

Merging Users

- How is this merge request link generated?

https://piazza.com/verify_email/
hf5cxrptw2a1vm/hf6rag7dikg5lb?preferred=yes

ALICEs *uid*

HASH code

Status

Merging Users

- How is this merge ~~request~~ Accept link generated?

https://piazza.com/verify_email/hf5cxrptw2a1vm/hf6rag7dikg5lb?verify=yes

ALICEs *uid* — HASH code — Status

Merging Users

Login

piazza

All Done :)

Your email address has been successfully linked to your account.
Click here to continue to your class.

Company
Our Story
Our Team
Our Investors
Press Releases
In the News
Jobs

Product
Why Piazza Works
Features
Product FAQ
Instructor FAQ
LMS Integration
Accessibility

Piazza In Action
Computer Science
Engineering
Physics
Math
Chemistry
Biology

Support
Help
Contact Us
Resources For Instructors

Links
Home
Blog
Mobile
Login Page

Find us on:

Our Terms have changed, you can review the Piazza Terms of Service.

Privacy Policy · Copyright Policy · Terms of Service · FERPA

Merging Users

- Can Alice (the adversary) predict this URL?

https://piazza.com/verify_email/
hf5cxrptw2a1vm/hf6rag7dikg5lb?verify=yes

ALICEs *uid* — HASH code — Status

Merging Users

- Can Alice (the adversary) predict this URL?

https://piazza.com/verify_email/
hf5cxrptw2a1vm/hf6rag7dikg5lb?verify=yes

ALICEs *uid* — HASH code — Status ✓

Merging Users

- Can Alice (the adversary) predict this URL?

https://piazza.com/verify_email/
hf5cxrptw2a1vm/hf6rag7dikg5lb?verify=yes

✓ **ALICE**s *uid* HASH code Status ✓

Merging Users

- Can Alice (the adversary) predict this URL?

https://piazza.com/verify_email/
hf5cxrptw2a1vm/hf6rag7dikg5lb?verify=yes

✓ **ALICE**s *uid*

HASH code ?

Status ✓

hf6rag7d

ikg5lb

Merging Users

- Can Alice (the adversary) predict this URL?

https://piazza.com/verify_email/
hf5cxrptw2a1vm/hf6rag7dikg5lb?verify=yes

✓ **ALICE**s *uid*

HASH code ?

Status ✓

hf6rag7d

ikg5lb

- Base 36 Encoding [a-z,0-9]
- Entropy: 36^8 => $O(2^{41})$ => 41 bits
- UNIX Epoch timestamp
  (milliseconds elapsed since 1 Jan, 1970 UTC)
- Predictable?
  - Ad-Hoc experimentation allowed me to narrow this down to a 1 second window
- Effective Entropy

  $O(2^{10})$ => 10 bits ☺

Merging Users

- Can Alice (the adversary) predict this URL?

https://piazza.com/verify_email/
hf5cxrptw2a1vm/hf6rag7dikg5lb?verify=yes

✓ **ALICE**s *uid*

HASH code ?

Status ✓

hf6rag7d

ikg5lb

- Base 36 Encoding [a-z,0-9]
- Entropy: 36^8 => $O(2^{41})$ => 41 bits
- UNIX Epoch timestamp
  (milliseconds elapsed since 1 Jan, 1970 UTC)
- Predictable?
  - Ad-Hoc experimentation allowed me to narrow this down to a 1 second window
- Effective Entropy

  $O(2^{10})$ => 10 bits ☺

- Base 36 Encoding [a-z,0-9]
- Entropy: 2^{32} => 32 bits
- 32-bit nonce
- Predictable?
  - Maybe…
- Effective Entropy

  2^{32} => 32 bits ☹

Merging Users

- Can Alice (the adversary) predict this URL?

https://piazza.com/verify_email/
hf5cxrptw2a1vm/hf6rag7dikg5lb?verify=yes

✓ **ALICE**s *uid*

HASH code ?

Status ✓

hf6rag7d

ikg5lb

- Base 36 Encoding [a-z,0-9]
- Entropy: 36^8 => $O(2^{41})$ => 41 bits
- UNIX Epoch timestamp
  (milliseconds elapsed since 1 Jan, 1970 UTC)
- Predictable?
  - Ad-Hoc experimentation allowed me to narrow this down to a 1 second window
- Effective Entropy

  $O(2^{10})$ => 10 bits ☺

- Base 36 Encoding [a-z,0-9]
- Entropy: 2^{32} => 32 bits
- 32-bit nonce
- Predictable?
  - Maybe…
- Effective Entropy

  2^{32} => 32 bits ☹

~ 42 bits Total Entropy

Merging Users

Merging Users

Miscalculations & Gross Oversights

- Brute Forcing a remote system is NOT the same as brute forcing hashes offline

Merging Users

Miscalculations & Gross Oversights

- Brute Forcing a remote system is NOT the same as brute forcing hashes offline
  - HTTP Requests take time… can't throw a GPU at it.
  - DOS mitigations
  - Detection!
  - Time to complete (being *very* generous)

Merging Users

Miscalculations & Gross Oversights

- Brute Forcing a remote system is NOT the same as brute forcing hashes offline
  - HTTP Requests take time… can't throw a GPU at it.
  - DOS mitigations
  - Detection!
  - Time to complete (being *very* generous)
    - (to hit every one)

Merging Users

Miscalculations & Gross Oversights

- Brute Forcing a remote system is NOT the same as brute forcing hashes offline
  - HTTP Requests take time… can't throw a GPU at it.
  - DOS mitigations
  - Detection!
  - Time to complete (being *very* generous)
    - (to hit every one)

Merging Users

Miscalculations & Gross Oversights

- Brute Forcing a remote system is NOT the same as brute forcing hashes offline
  - HTTP Requests take time… can't throw a GPU at it.
  - DOS mitigations
  - Detection!
  - Time to complete (being *very* generous)
    - (to hit every one)
    - (to have a reasonable chance of success)

Merging Users

All hope is not lost

- Perhaps knowing the time the request was generated will make the nonce more predictable?

Merging Users

All hope is not lost

- Perhaps knowing the time the request was generated will make the nonce more predictable?
  - Poor nonce generation by Piazza developers?
  - Weakness in Java Random API implementation?

Merging Users

All hope is not lost

- Perhaps knowing the time the request was generated will make the nonce more predictable?
  - Poor nonce generation by Piazza developers?
  - Weakness in Java Random API implementation?
- Perhaps the nonce is not really random at all, or is at least influenced by other factors

  » *uid (sender and receiver), nid, aid*

Merging Users

All hope is not lost

- Perhaps knowing the time the request was generated will make the nonce more predictable?
  - Poor nonce generation by Piazza developers?
  - Weakness in Java Random API implementation?
- Perhaps the nonce is not really random at all, or is at least influenced by other factors

  » *uid (sender and receiver), nid, aid*

  - Multiple merge requests to/from same parties will result in identical nonce values
  - More research is needed regarding nonce generation

Merging Users

All hope is not lost (Part II)

- Why just focus on the hash code? What if we could tamper with how the merge request is delivered?

Merging Users

All hope is not lost (Part II)

- Why just focus on the hash code? What if we could tamper with how the merge request is delivered?
- Piazza uses *JavaMail* to deliver it's emails

Merging Users


```
        new0=
DomainKey-Signature: a=rsa-sha1; c=nofws; d=piazza.com; h=from:to
        :subject:mime-version:content-type; q=dns; s=smtpapi; b=WWyHctM3
        mA3dgoNNH17NDa+S0Aq8XuqVhYKHPs2bAUfZm9NjdA1o9978Xp7bLaSd3TeA7bdb
        HwHmb52WZU0+U3Zw4PhrMJ8cr0BUTydNaqFa56iKEU2GhQadRMiFJhUKEJuHjYzk
        rGD5o/HPuLPPG1Gz2RxFayDRoFthAhaQPKY=
Received: by 10.36.109.179 with SMTP id mf48.26340.5169ECEA5
        Sat, 13 Apr 2013 23:40:26 +0000 (UTC)
Received: from smtp.sendgrid.net (ec2-50-19-144-23.compute-1.amazonaws.com [50.19.144.23])
        by mi20 (SG) with ESMTP id 5169ece9.be6.10ae97
        for <the.l.i.z.ard.kingami@gmail.com>; Sat, 13 Apr 2013 18:40:25 -0500 (CST)
Date: Sat, 13 Apr 2013 23:40:24 +0000 (UTC)
From: Piazza Team <no-reply@piazza.com>
To: the.l.i.z.ard.kingami@gmail.com
Message-ID: <2035347102.2061.1365896425092.JavaMail.ec2-user@appserver03>
Subject: Confirm Your Email Address for Piazza
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="----=_Part_2059_904220434.1365896424980"
X-SG-EID:
qBvDszCdXxEIaeo11IAFw8eTRGvPi5zYMD32tZZX+MuZbu1xNi3YwI4F6MttFBjCEMkuE8mTzWvfNB2W/VsKUtFcupBiCaFKfLzW64ERsK/zCANV5391js9cYXSYssuS
YCSCLUdqUx+4NAEfJq9O/A==

------=_Part_2059_904220434.1365896424980
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 7bit

Hello!

We received a request to merge the.l.i.z.ard.kingami@gmail.com to your mbclark@ucalgary.ca account on Piazza.

To confirm this request, follow this link: https://piazza.com/verify_email/gxbbchjnaog4pb/hfhfgpu45zo2je?preferred=no

If you did not send this request, you can safely ignore it. Contact us at team@piazza.com or 1-800-818-4124 if you need help.

Thanks,
The Piazza Team
```

Merging Users

All hope is not lost (Part II)

- Why just focus on the hash code? What if we could tamper with how the merge request is delivered?
- Piazza uses *JavaMail* to deliver it's email
  - Attempts to trick the service into delivering to multiple recipients failed
    - target@gmail.com; evil@gmail.com
    - target@gmail.com, evil@gmail.com
  - Functionality not a part of *JavaMail*
  - No CVEs found for allowing this

Merging Accounts

Assuming we could somehow abuse our knowledge of a user's email address…

…how would one expect to find out the email address of other users?

Information Leakage

Information Leakage

- During analysis of the client-side JavaScript code two interesting objects sitting in the DOM were discovered

Information Leakage

- During analysis of the client-side javascript code two interesting objects sitting in the DOM were discovered
  - PA.user
    - Local cache of user information
    - Populated from the response to an API call
    - Cache is used to minimize network traffic

Information Leakage

Elements | Resources | Network | Sources | Timeline | Profiles | Audits | Console

```
> PA.user
  ▼ Object {networks: Array[2], sid: "hfjqvp7qfxb4k9", new_questions: Object, last_network: "hf4zsf2w7jv5ri", last_content: Object…}
      activated: 1365245862603
    ▶ can_admin: Object
      can_anonymize: false
    ▶ config: Object
      email: "t.h.is.isadu.m.m.e@gmail.com"
    ▶ emails: Array[3]
    ▶ facebook: Object
      facebook_id: undefined
      feed_prefetch: null
      id: "hf5cxwe5d6r2fu"
      is_admin: false
      is_public: false
    ▶ last_content: Object
      last_network: "hf4zsf2w7jv5ri"
      name: "Dumme Daves Uncle"
    ▶ networks: Array[2]
    ▶ new_questions: Object
      photo: "1366036651_35.png"
      photo_original: ""
    ▶ profile: Object
      sid: "hfjqvp7qfxb4k9"
    ▶ __proto__: Object
>
```

Information Leakage

The more interesting of the two DOM objects

- PA.Users[uid]
  - Local cache of other users *known* to the logged-in user
  - *Known*: any other user who has posted something you have viewed, or your browser has rendered their username on the screen
  - Students are always *known* to their professors, and Professors to their Students
  - Similarly, is populated via a series of API calls
    - Network.get_users([uid],nid)

Information Leakage

Elements | Resources | Network | Sources | Timeline | Profiles | Audits | Console

```
> PA.users
  ▼ Object {hf4zt6r3xof536: Object, gxbbchjnaog4pb: Object, hf5cxwe5d6r2fu: Object, hf5cxzad7zv2qd: Object, hf5cy1w8zzp32m: Object}
    ▼ gxbbchjnaog4pb: Object
        admin: true
        admin_permission: 15
        email: "mbclark@ucalgary.ca"
        facebook_id: null
        id: "gxbbchjnaog4pb"
        name: "Michael Clark"
        photo: "1327296898.png"
        role: "Professor"
        us: false
      ▶ __proto__: Object
    ▶ hf4zt6r3xof536: Object
    ▼ hf5cxwe5d6r2fu: Object
        id: "hf5cxwe5d6r2fu"
        name: "Dumme Daves Uncle"
        photo: "1366036651_35.png"
      ▶ __proto__: Object
    ▼ hf5cxzad7zv2qd: Object
        admin: false
        facebook_id: null
        id: "hf5cxzad7zv2qd"
        name: "Dumme Dave Aunt"
        photo: null
        role: "student"
        us: false
      ▶ __proto__: Object
    ▼ hf5cy1w8zzp32m: Object
        admin: false
        facebook_id: "123456"
        id: "hf5cy1w8zzp32m"
        name: "Dumme Dave Sister"
        photo: null
        role: ""
        us: false
      ▶ __proto__: Object
    ▶ __proto__: Object
> |
```

Information Leakage

- However, you can make the API call directly to get user information
  - Network.get_users([uid_array],nid)
- Furthermore, if you omit *nid*, the API will return successfully!!!
- What does this mean?
  - We can retrieve user info on ANY user, as long as we know their *uid*, regardless of if they are in our network

Information Leakage

- Enumerating Users
  - In theory, we could enumerate every user on Piazza
  - In practice, we may only want to creep on our fellow classmates

Information Leakage

- Enumerating Users
  - In theory, we could enumerate every user on Piazza
  - In practice, we may only want to creep on our fellow classmates
  - How?
    - Recall: The *nid* (which is common knowledge) is incorporates a timestamp…. So does the user id.
    - Also, *uid*s are unique, in that the second half is sequential
    - Start making API calls based from when the class was created…
    - or better yet, if the instructor did a batch add, begin exploring from our *uid* upwards and downwards in value

Information Leakage

- What else can we find out?

Information Leakage

- What else can we find out?
  - Facebook user ids

Information Leakage

Elements | Resources | Network | Sources | Timeline | Profiles | Audits | Console

```
> PA.users
  ▼Object {hf4zt6r3xof536: Object, gxbbchjnaog4pb: Object, hf5cxwe5d6r2fu: Object, hf5cxzad7zv2qd: Object, hf5cy1w8zzp32m: Object}
    ▼gxbbchjnaog4pb: Object
       admin: true
       admin_permission: 15
       email: "mbclark@ucalgary.ca"
       facebook_id: null
       id: "gxbbchjnaog4pb"
       name: "Michael Clark"
       photo: "1327296898.png"
       role: "Professor"
       us: false
     ▶__proto__: Object
    ▶hf4zt6r3xof536: Object
    ▼hf5cxwe5d6r2fu: Object
       id: "hf5cxwe5d6r2fu"
       name: "Dumme Daves Uncle"
       photo: "1366036651_35.png"
     ▶__proto__: Object
    ▼hf5cxzad7zv2qd: Object
       admin: false
       facebook_id: null
       id: "hf5cxzad7zv2qd"
       name: "Dumme Dave Aunt"
       photo: null
       role: "student"
       us: false
     ▶__proto__: Object
    ▼hf5cy1w8zzp32m: Object
       admin: false
       facebook_id: "123456"
       id: "hf5cy1w8zzp32m"
       name: "Dumme Dave Sister"
       photo: null
       role: ""
       us: false
     ▶__proto__: Object
    ▶__proto__: Object
>
```

Information Leakage

- What else can we find out?
  - Facebook user ids
    - One would not expect that by using the Facebook connect feature (an authentication convenience feature), that other Piazza users would be able to link you to your Facebook

Information Leakage

- What else can we find out?
  - Facebook user ids
    - One would not expect that by using the Facebook connect feature (an authentication convenience feature), that other Piazza users would be able to link you to your Facebook
    - What if you are bad-mouthing our Professor?

Information Leakage

- What else can we find out?
  - Facebook user ids
    - One would not expect that by using the Facebook connect feature (an authentication convenience feature), that other Piazza users would be able to link you to your Facebook
    - What if you are bad-mouthing our Professor?
    - What if your Professor is bad-mouthing you?

Information Leakage

- More confidential information?
  - How about private communications between students and instructors?

Information Leakage

Console HTML CSS Script DOM Net Cookies

Clear Persist All HTML CSS JS XHR Images Flash Media

```
resource_sections
  [ Object { title="Homework", date_title="Due Date", visibility=true, more… }, Object { title="Homework Solutions", visibility=false, name="homework_solutions", more… }, Object { title="Lecture Notes", date_title="Lecture Date", visibility=true, more… }, Object { title="General Resources", visibility=true, name="general", more… } ]
seen_message
  [ "first_day", "tips_tricks", "startclass-3d-after-lt50", "startclass-7d-after-lt80" ]
tips_tricks_nr
  2
wishlist
  Object { requesting_users=[1], requests=[2] }
  requesting_users
    [ "h6xur6k0y945ng" ]
  requests
    [ Object { to=[1], uid="h6xur6k0y945ng", text="Dear Dr. Mohassel,\n\nI c...C 526 student ", more… }, Object { to=[1], uid="h6xur6k0y945ng", text="Dear Dr. Mohassel,\n\nI c...C 526 student ", more… } ]
    0
      Object { to=[1], uid="h6xur6k0y945ng", text="Dear Dr. Mohassel,\n\nI c...C 526 student ", more… }
      to
        [ "pmohasse@cpsc.ucalgary.ca" ]
      uid
        "h6xur6k0y945ng"
      text
        "Dear Dr. Mohassel,

        I created CPSC 526 on piazza.com due to the discussion in class. If you would like to, you could continue the process of registering this class.

        It only takes two minutes to set up a class:
        https://piazza.com/?nid=hbsqin344st6in

        Kind regards,
        Adrian Cristea, CPSC 526 student "
      subject
        "Can we use Piazza for CPSC 526?"
      date
        "2013-01-11T03:37:37Z"
    1
      Object { to=[1], uid="h6xur6k0y945ng", text="Dear Dr. Mohassel,\n\nI c...C 526 student ", more… }
      to
        [ "pmohasse@cpsc.ucalgary.ca" ]
      uid
        "h6xur6k0y945ng"
      text
        "Dear Dr. Mohassel,\n\nI c...C 526 student "
      subject
        "Can we use Piazza for CPSC 526?"
      date
        "2013-01-11T03:37:48Z"
syllabus
  ""
enrollment
  25
course_description
  ""
profs
  [ Object { id="hbssul5ict248e", email="pmohasse@cpsc.ucalgary.ca", admin=true, more… } ]
course_number
  "CPSC 526"
status
  "active"
```

**Used with the *explicit* written consent of both parties

Information Leakage

- Finally, you may ask:

  What good is this if you must provide a valid university email to register (or at least the Professor does)?

  Doesn't this guarantee attribution, and thus afford some sort of protection against a malicious user exploiting these vulnerabilities?

Information Leakage

piazza

Log In

Hard Knocks

(change school)

Are you a professor?
Click here to create & join classes

Welcome to Piazza!

Piazza is a free platform for instructors to efficiently manage class Q&A. Students can post questions and collaborate to edit responses to these questions. Instructors can also answer questions, endorse student answers, and edit or delete any posted content.

Piazza is designed to simulate real class discussion. It aims to get high quality answers to difficult questions, fast!

The name Piazza comes from the Italian word for plaza--a common city square where people can come together to share knowledge and ideas. We strive to recreate that communal atmosphere among students and instructors.

Your Spring 2013 Classes:

Class 1: 101: Keepin it Real (new class created!)
✔ **Join as:** Student TA Professor

Class 2:

Class 3:

Class 4:

Class 5:

Add Another Class

Add Classes

Unable to sign up? Email us at team@piazza.com and we'll help you get started!

Information Leakage

```
Received: by 10.42.80.142 with SMTP id filter-076.18359.5167DD5A1
        Fri, 12 Apr 2013 10:09:30 +0000 (UTC)
Received: from smtp.sendgrid.net (ec2-54-242-110-184.compute-1.amazonaws.com [54.242.110.184])
        by mi21 (SG) with ESMTP id 5167dd5a.4c79.141e09
        for <the.li.zard.king.a.m.i@gmail.com>; Fri, 12 Apr 2013 05:09:30 -0500 (CST)
Date: Fri, 12 Apr 2013 10:09:30 +0000 (UTC)
From: Piazza Team <no-reply@piazza.com>
To: the.li.zard.king.a.m.i@gmail.com
Message-ID: <1174509471.6293.1365761370347.JavaMail.ec2-user@ip-10-30-135-56>
Subject: Access code for 101
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="----=_Part_6292_1901334633.1365761370228"
X-SG-EID: xRK9Rr6FjcReaZcp2Fj7/hq7O3Qmd3xcVZhhgDTYbmMNeIO8mVb8EqaFWGRRCXGIt53Hmm+D/d5aqBwT2AccNZp+OFlViE709wtdky18YiKCSKEfp//eJj5Y1yxT3bt7K4aAOwe15g1DQ3Hy8bOCiQ==

------=_Part_6292_1901334633.1365761370228
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 7bit

The access code for your class is 101

Please share this code with anyone who will be enrolling themselves in your class: http://piazza.com/hard_knocks/spring2013/101

You can change your code at any time through your "Manage Class" page.

Thanks,
The Piazza Team
--
Contact us at team@piazza.com

------=_Part_6292_1901334633.1365761370228
MIME-Version: 1.0
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: 7bit

<html>
The access code for your class is <b>101</b>
<br><br>
Please share this code with anyone who will be enrolling themselves in your class: http://piazza.com/hard_knocks/spring2013/101
<br><br>
You can change your code at any time through your "Manage Class" page.
<br>
<br>
Thanks,<br>
The Piazza Team<br>
--<br>
Contact us at team@piazza.com<br>
</html>
------=_Part_6292_1901334633.1365761370228--
```

Information Leakage

- Finally, you may ask:

  What good is this if you must provide a valid university email to register (or at least the Professor does)?

  Doesn't this guarantee attribution, and thus afford some sort of protection against a malicious user exploiting these vulnerabilities?

- No!
  - Anyone can create a fictitious school, using a throwaway email address.

Vendor Response

Vendor Response

- Vendor was contacted through a built in Bug Reporting widget within the Piazza web interface.

Vendor Response?

Nothing

- Vendor was again contacted, this time through a different ‘contact us’ link on the web site portal.

Still awaiting a response…
Actually heard back this morning Monday April 15, 2013.

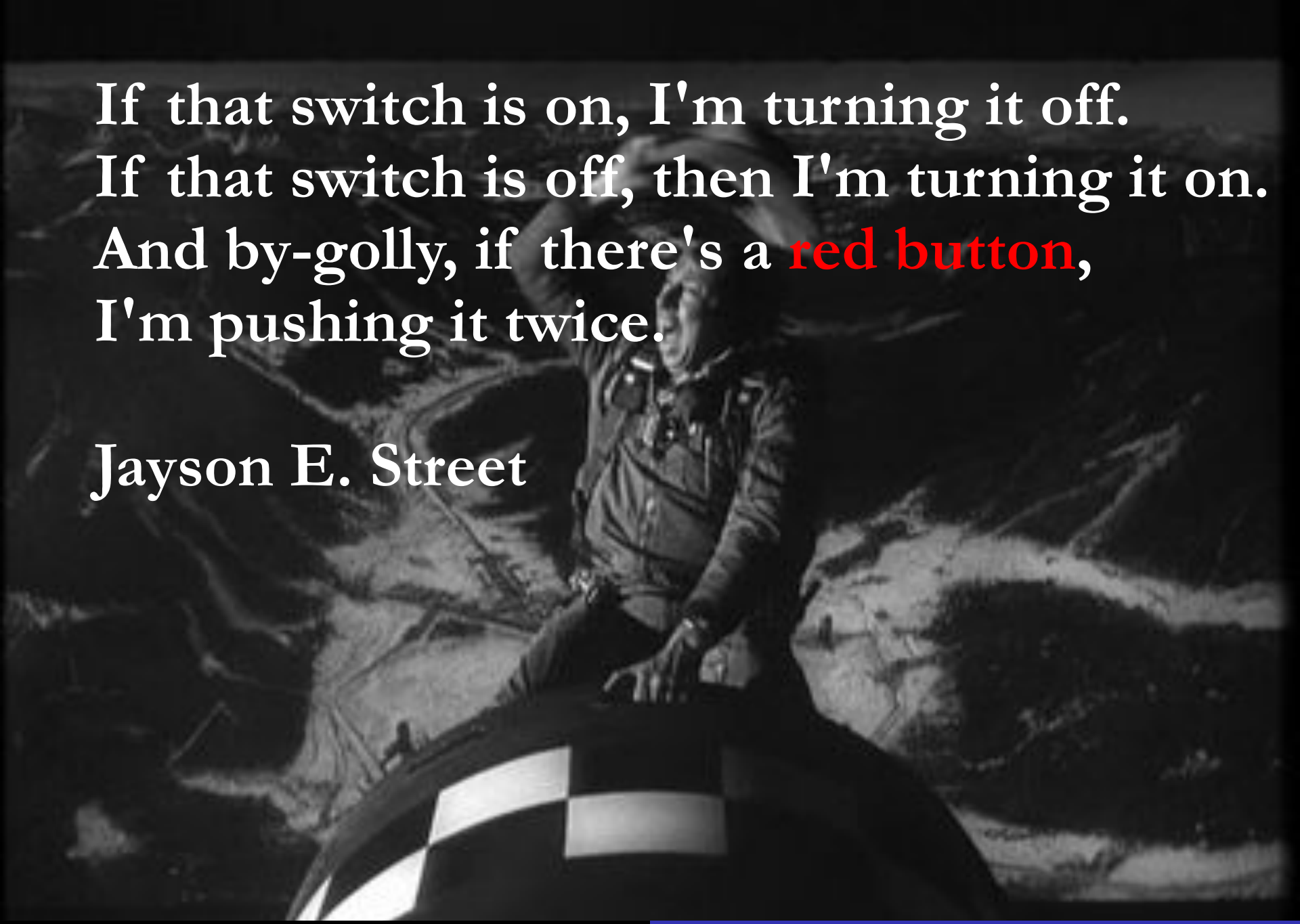
Conclusion

- Didn’t get root
- Didn’t drop a shell (sorry Alex)

BUT

- I was able to use the existing API to do some really cool things
- Systems are complex
- APIs present a large attack surface
- Breaking the API can be fun and rewarding

  (for both yourself and the vendor… free QA!)



If that switch is on, I'm turning it off.
If that switch is off, then I'm turning it on.
And by-golly, if there's a red button,
I'm pushing it twice.

Jayson E. Street


THE END

Questions?

Obligatory “Don’t-Sue-Me” Blurb