B@BEL: Leveraging Email Delivery for Spam Mitigation

Gianluca Stringhini, Manuel Egele, Apostolis Zarras,
Thorsten Holz, Christopher Kruegel,
and Giovanni Vigna

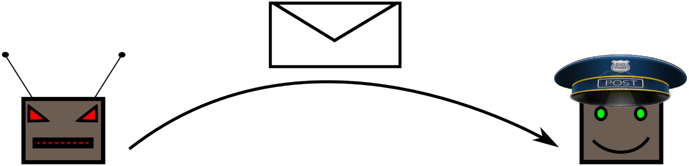
USENIX Security Symposium

August 8, 2012

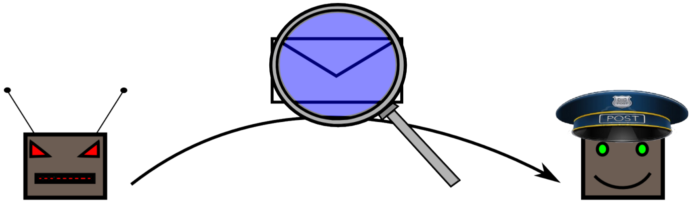
Spam is a big problem

- Wealthy economy behind spam
- 77% of emails are spam
- Botnets responsible for 85% of spam

Traditional spam detection

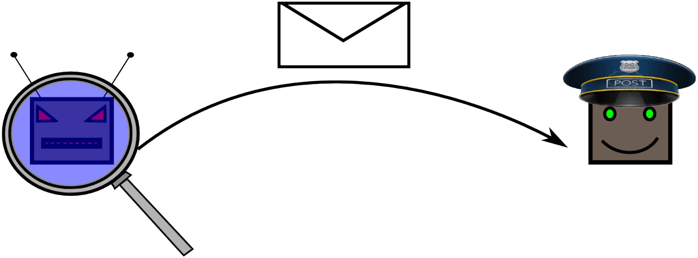


Traditional spam detection



Content analysis (What?)

Traditional spam detection

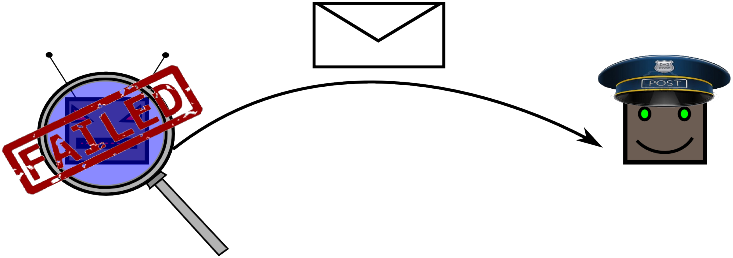


Origin analysis (Who?)

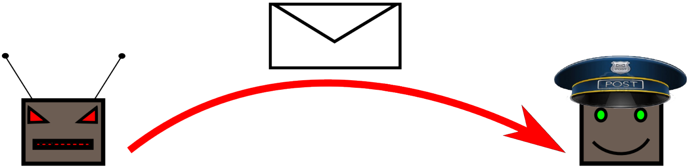
Existing methods have problems

Existing methods have problems

Existing methods have problems



Our approach



The way clients interact with SMTP servers (How?)

Two instances of our approach

- SMTP dialects
- Feedback manipulation

Outline of the talk

Techniques overview ←

System design

Evaluation

Limitations

First technique: SMTP dialects

The SMTP protocol

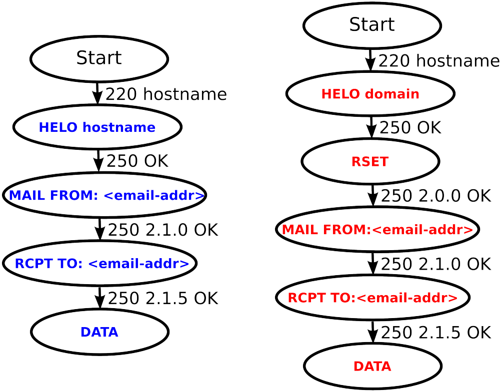
Server: 220 server
Client: HELO example.com
Server: 250 OK
Client: MAIL FROM:<me@example.com>
Server: 250 2.1.0 OK
Client: RCPT TO:<you@example.com>
Server: 250 2.1.5 OK
Client: DATA

"Be conservative in what you send, but liberal in what you accept"
(Postel's Law)

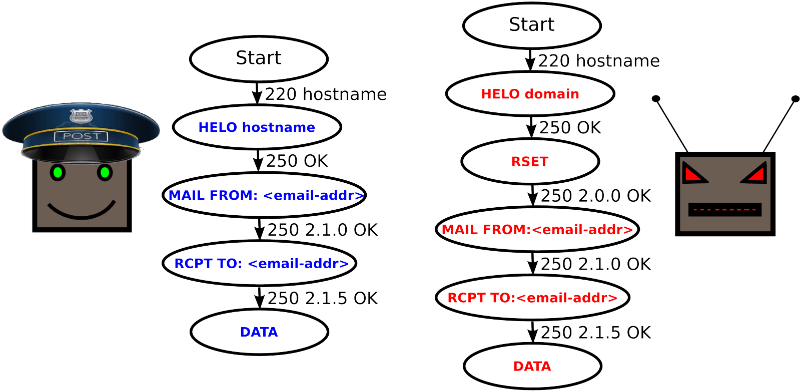
SMTP dialects

SMTP dialects

SMTP dialects



SMTP dialects



What can we use dialects for?

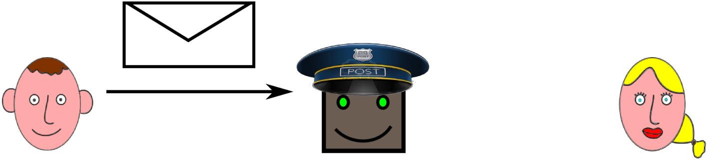
- Spam detection
- Malware classification

Second technique: feedback manipulation

Feedback to emails

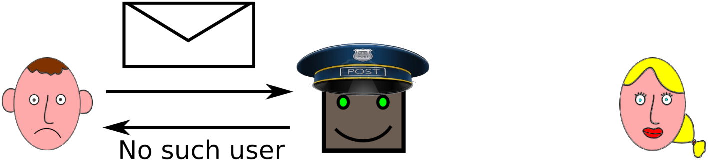
Feedback to emails

Feedback to emails

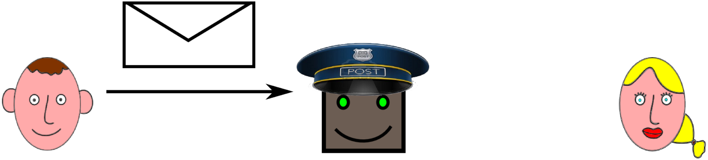


Feedback to emails

Feedback to emails

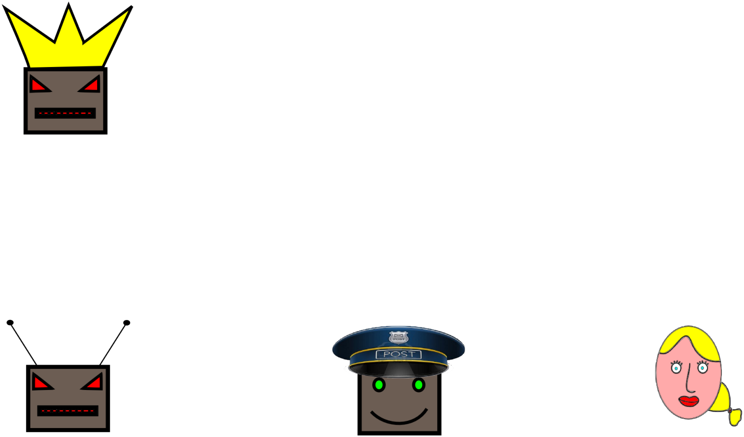


Feedback to emails

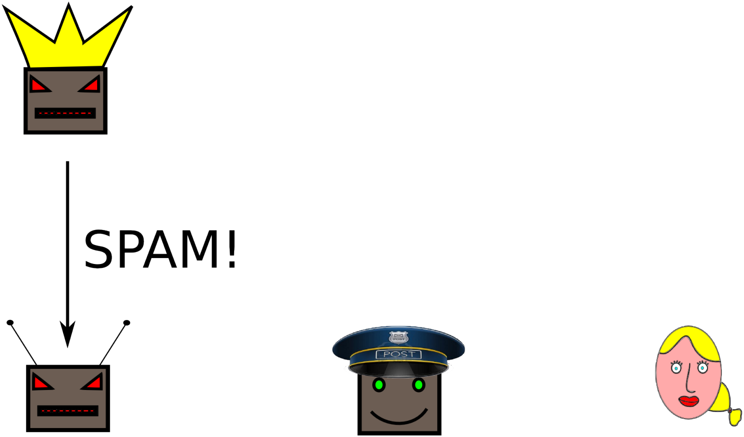


Feedback is important

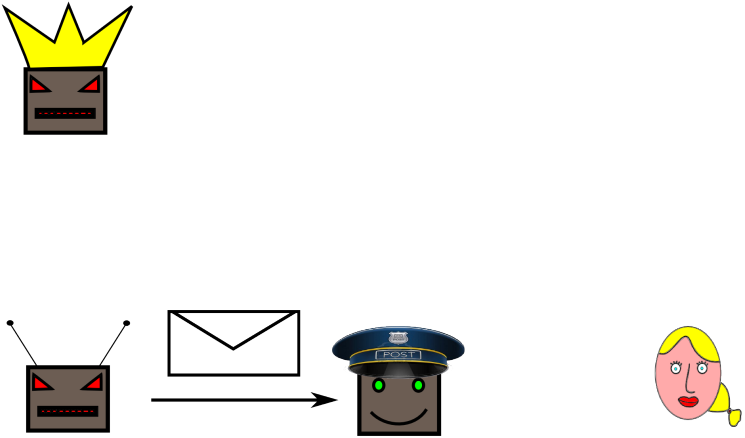
Botnets use this feedback too



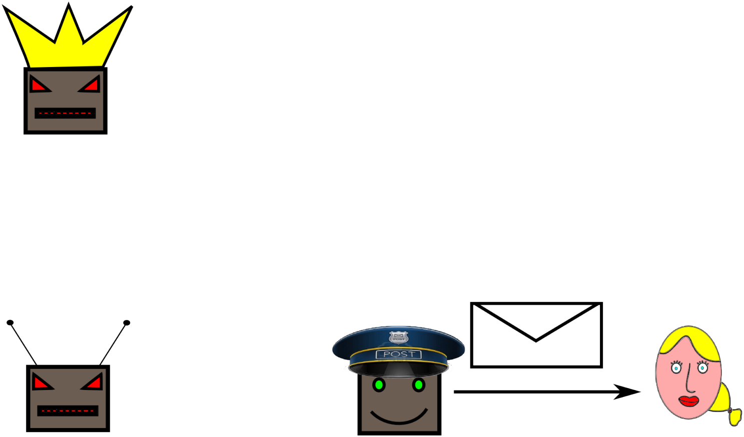
Botnets use this feedback too



Botnets use this feedback too

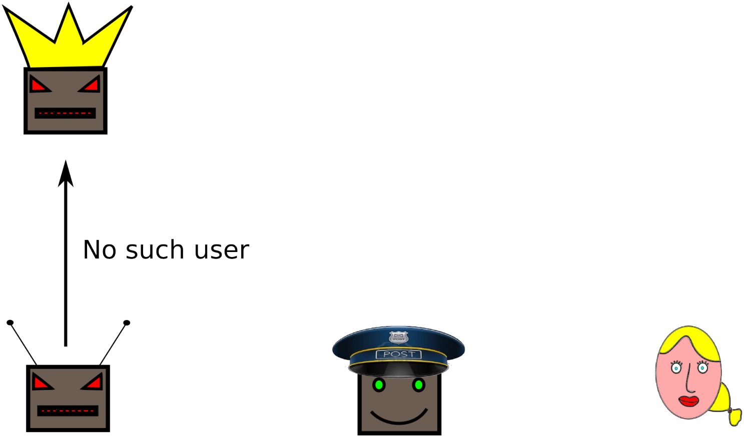


Botnets use this feedback too



Botnets use this feedback too

Botnets use this feedback too



How important is feedback?

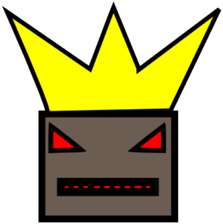
Previous research

- Succesful botnets are using bot feedback
- Cutwail: 35% of the email addresses were nonexistent

What if we gave wrong feedback?

What should the botmaster do?

THE COMPUTER SECURITY GROUP AT UCSB



Lose-lose situation

- Accept feedback
- Discard feedback

Outline of the talk

Techniques overview

System design ←

Evaluation

Limitations

A typical SMTP conversation

Server: 220 server
Client: HELO example.com
Server: 250 OK
Client: MAIL FROM:<me@example.com>
Server: 250 2.1.0 OK
Client: RCPT TO:<you@example.com>
Server: 250 2.1.5 OK
Client: DATA

Dialects as state machines

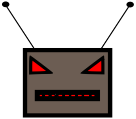
$$\mathbf{D} = < \Sigma, S, s_0, T, F_g, F_b >$$

- Σ: input alphabet
- S: set of states
- s_0: initial state
- T: transitions
- F_g: "good" final states
- F_b: "bad" final states

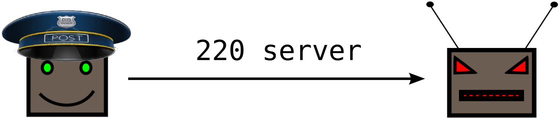
Three phases

- Learning SMTP dialects
- Building a decision model
- Making a decision

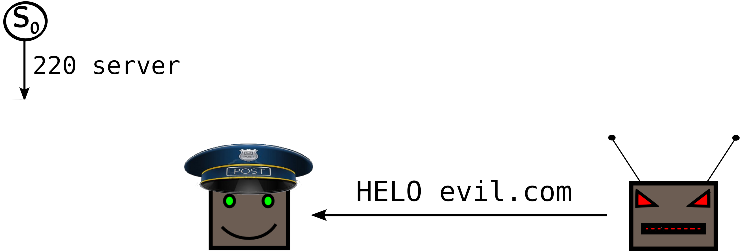
Learning SMTP dialects



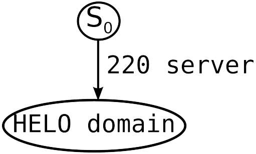
Learning SMTP dialects

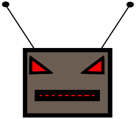


Learning SMTP dialects

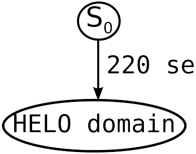


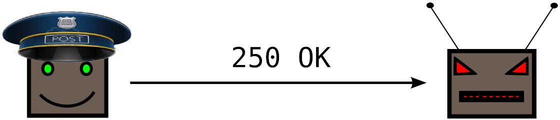
Learning SMTP dialects



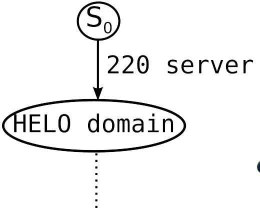


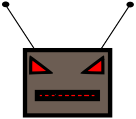
Learning SMTP dialects





Learning SMTP dialects





Collecting SMTP conversations

Passive observation

Two dialects might look the same!

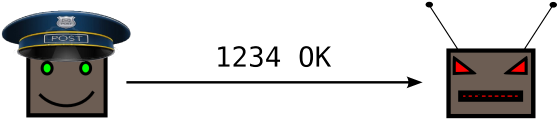
Active probing

Send incorrect replies, error messages, ...

Active probing

Out-of-order replies

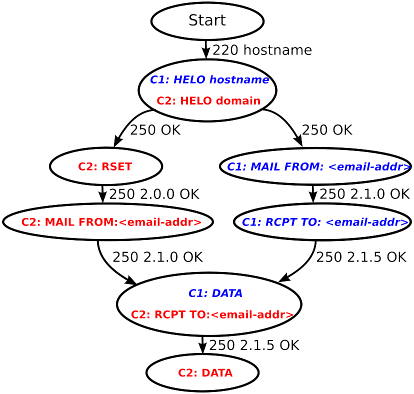
Active probing



Incorrect replies

Building a decision model

Building a decision model



Making a decision

Passive matching

Detect dialects by observing conversations

Active probing

Send specific replies to “expose” differences

Outline of the talk

Techniques overview

System design

Evaluation ←

Limitations

Dialects for classification

Our experiment

- 13 legitimate MUAs and MTAs
- 91 distinct malware samples
- We performed active probing (228 variations)

Results

- Legitimate and malicious dialects are distinct
- Malware families all speak different dialects
- Better classification than AV labels

Dialects for spam mitigation

Our experiment

621,919 SMTP conversations

Results

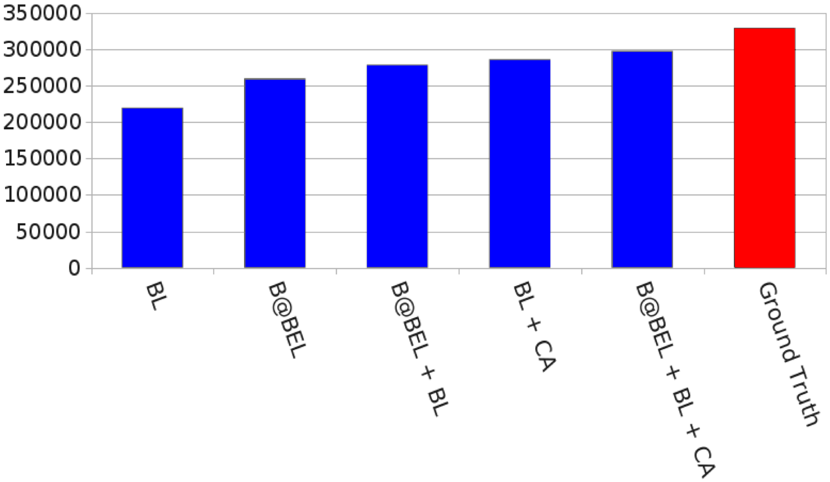
- 260,074 as bots
- 218,675 as legitimate clients
- 143,170 no decision

How accurate is B@BEL?

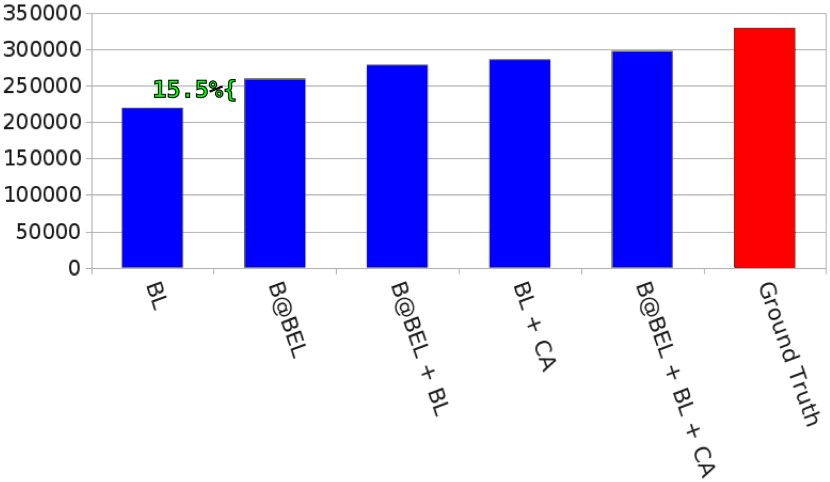
- 0.67% false positives
- 21% false negatives

B@BEL detects email engines, not content!

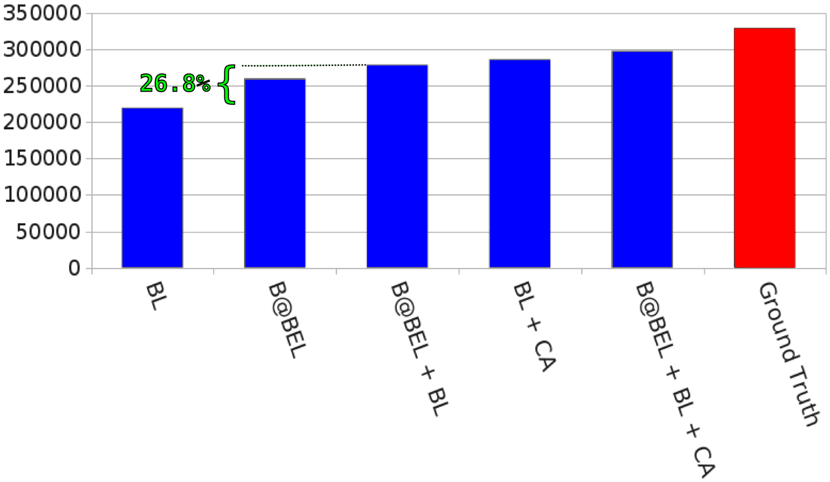
Is it worth it?



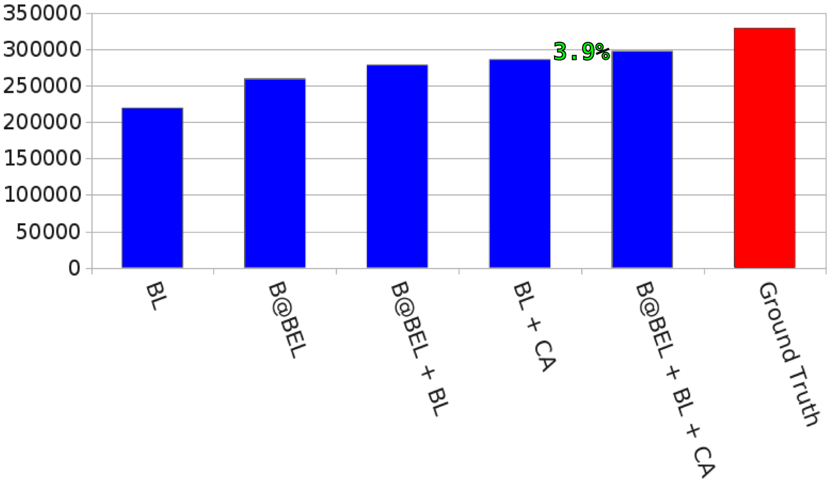
Is it worth it?



Is it worth it?



Is it worth it?

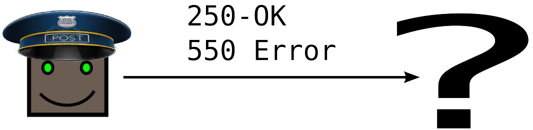


How about those 143,170 emails?

?

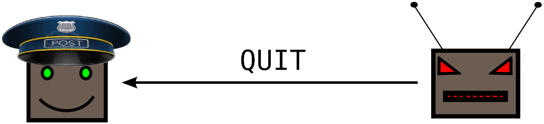
Disambiguation is always possible with one reply

How about those 143,170 emails?



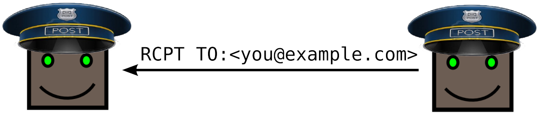
Disambiguation is always possible with one reply

How about those 143,170 emails?



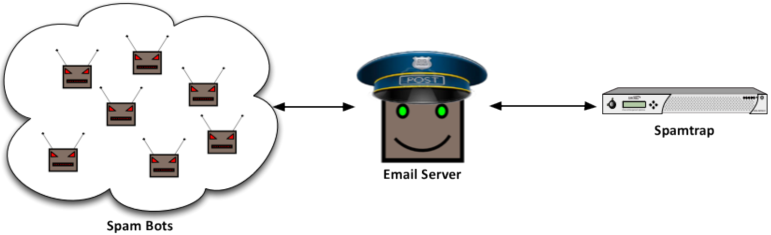
Disambiguation is always possible with one reply

How about those 143,170 emails?



Disambiguation is always possible with one reply

Giving wrong feedback – Evaluation



Our experiment

- 32 malware samples
- Sinkholed the emails sent by the bots
- Looked at the effect on our spam trap

Giving wrong feedback – Evaluation

Results

- Sent feedback to 29 campaigns — 2.8M emails
- For 5 of them the technique worked
- 19% of the total number of emails!

Outline of the talk

Techniques overview

System design

Evaluation

Limitations ←

Limitations

Evading dialects detection

- Implement a “faithful” SMTP engine

  Performance penalty!

- Force spammers to look like client X

  Easier to detect by previous work

Limitations

Evading feedback manipulation

Lose-lose situation for the botmaster

Conclusions

THE COMPUTER SECURITY GROUP AT UCSB

- B@BEL looks at how SMTP engines interact with mailservers
  - SMTP dialects
  - Feedback manipulation
- Valuable tool to aid spam mitigation
- Raises the bar for botmasters

Questions?

email: gianluca@cs.ucsb.edu
twitter: @gianlucaSB