

Fake Accounts: Theory and Practice

Devin Gaffney
International Persuasion Machines
@dgaff_



Who is this Person?

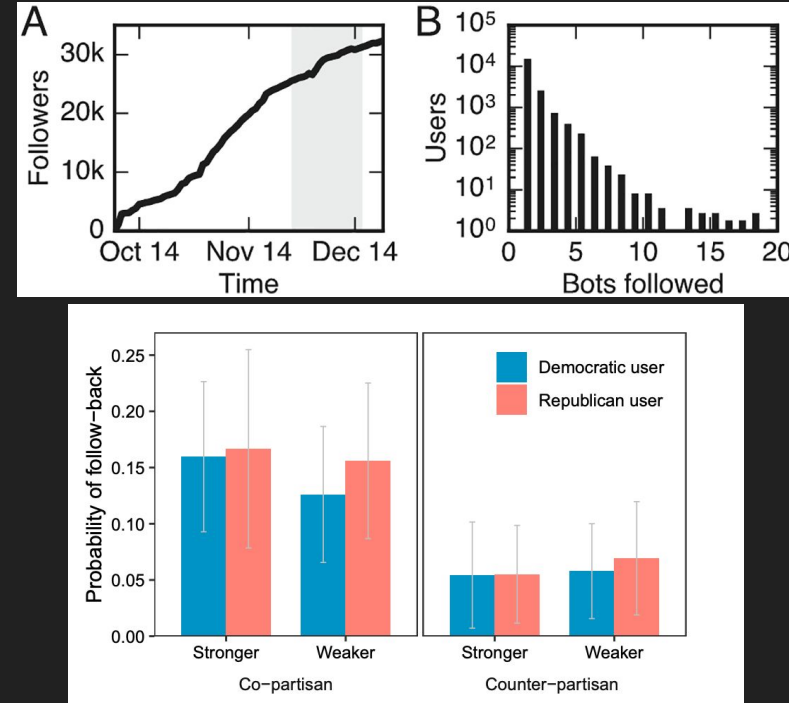
I'm Devin!

- 2008-2010 Bennington College, 140kit
- 2011-2012 OII, SocialBots, Twitter Geography
- 2013-2014 Lead @ Little Bird
- 2014-2018 Northeastern, Reddit Contagion
- 2020-2021 IPM
- Has Dogs



Do Bots Even Work?

- Yes, But Scant IRB-Backed Evidence
 - 30k Bots Built for Single Paper
 - 15% Follow-Back on Co-Partisans
 - Hard to Ethically Experiment
 - Detection Work Insufficient
- Presence Alone Degrades Platforms
- Bot Discourse Creates Incentives For Bot Creation
- Bots As Rhetorical Foil Regardless of Reality



Targets of Bot Builders

- On-Platform Users
 - Encourage Arguments ([Twitter](#))
 - Buy/Sell Influence ([Twitter](#))
- Off-Platform Users
 - Entice in-migration ([Ashley Madison](#))
- On-Platform Algorithms
 - Trending Topics ([Twitter](#))
 - Suggested Users/Products ([Amazon](#))
- Partnership Programs
 - View/Click Fraud ([YouTube](#))
- Anywhere where Incentives > Costs (Political|Financial)

Tools of Bot Builders

- Generation Procedures

- [Demographic Mimicry](#)
- [Photographic Mimicry](#)
- Adversarial Generation -
Update Generation Based on
Successful/Failed Attempts

- Interaction Interfaces

- [Python requests library](#)
- [Selenium](#)
- [Selenium Stealth](#)

- Network Interfaces

- [ExpressVPN](#) - VPN Masking
- [Luminati](#) - Proxy Masking
- [Hotspots](#) - Cell Network

Masking

- Grey Market APIs

- [2Captcha](#) - Break Captchas
- [Zadarma](#) - Buy/Access Phone
Numbers
- [Mailslurp](#) - Buy/Access Email
Addresses

- Reverse Engineering Target
Platforms

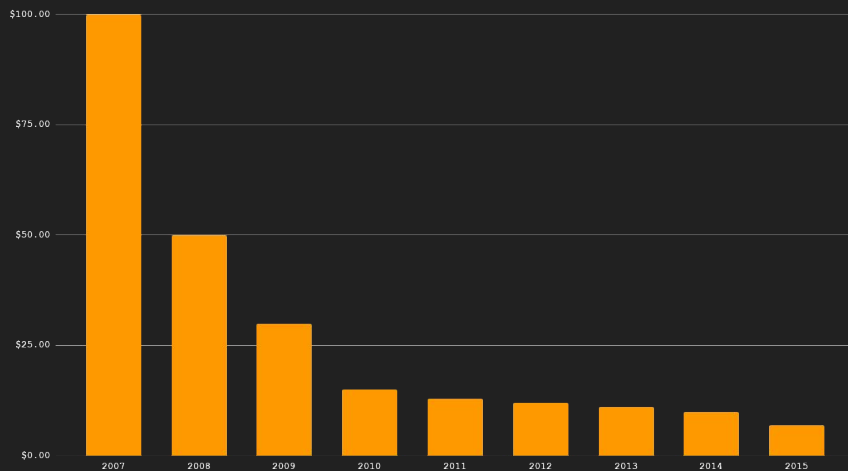
- [Network Inspector](#) - Investigate
Server Interactions
- [HTTP Echo Servers](#) - Test Outbound
Server Interactions
- [IfConfig.me](#) - Test Outbound IP
Environment



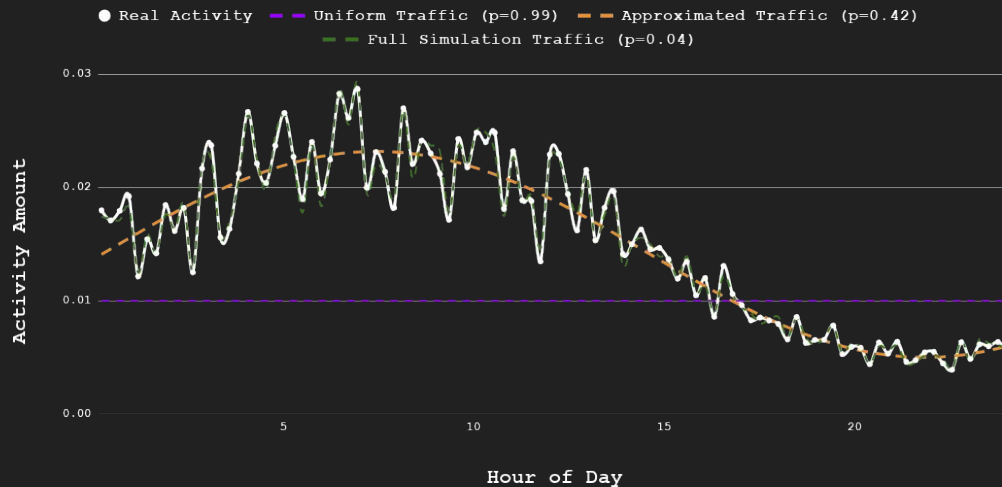
Techniques of Bot Builders

Aging

Aged Account Cost, AccFarm.com



Mimicking



Practical Demonstration – Demo App

SAMPLE APP

[Home](#) [Help](#) [Log In](#)

Sign up

Name

Email

Password

Confirmation

Create my account

The Ruby on Rails Tutorial by Michael Hartl

[About](#) [Contact](#) [News](#)

```
--- !ruby/object:ActionController::Parameters
parameters: !ruby/hash: ActiveSupport::HashWithIndifferentAccess
  controller: users
  action: new
  permitted: false
```



Practical Demonstration - Direct HTTP

```
import sys
import requests
from bs4 import BeautifulSoup
name = sys.argv[1]
email = sys.argv[2]
password = sys.argv[3]

page_load = requests.get("http://demo-rails-app-testing.ipm-corporation.com/signup")
cookies = dict(page_load.cookies)
soup = BeautifulSoup(page_load.text, 'html.parser')
auth_token = soup.findAll(attrs={"name" : "authenticity_token"})[0]['value']

data = {
    'utf8': '\u2713',
    'authenticity_token': auth_token,
    'user[name]': name,
    'user[email]': email,
    'user[password]': password,
    'user[password_confirmation]': password,
    'commit': 'Create my account'
}

response = requests.post(
    'http://demo-rails-app-testing.ipm-corporation.com/users',
    cookies=cookies,
    data=data,
    verify=False
)
```



Practical Demonstration – Direct HTTP, Fake User Agent

```
import sys
import requests
from bs4 import BeautifulSoup
name = sys.argv[1]
email = sys.argv[2]
password = sys.argv[3]

page_load = requests.get("http://demo-rails-app-testing.ipm-corporation.com/signup")
cookies = dict(page_load.cookies)
soup = BeautifulSoup(page_load.text, 'html.parser')
auth_token = soup.findAll(attrs={"name" : "authenticity_token"})[0]['value']

headers = {
    'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36'
}
data = {
    'utf8': '\u2713',
    'authenticity_token': auth_token,
    'user[name]': name,
    'user[email]': email,
    'user[password]': password,
    'user[password_confirmation]': password,
    'commit': 'Create my account'
}

response = requests.post(
    'http://demo-rails-app-testing.ipm-corporation.com/users?block_bot_useragents=true',
    cookies=cookies,
    data=data,
    headers=headers,
    verify=False
)
```



Practical Demonstration - Selenium

```
import sys
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
name = sys.argv[1]
email = sys.argv[2]
password = sys.argv[3]

driver = webdriver.Firefox()
driver.get("http://demo-rails-app-testing.ipm-corporation.com/signup")
element = driver.find_element_by_id("user_name")
element.send_keys(name)
element = driver.find_element_by_id("user_email")
element.send_keys(email)
element = driver.find_element_by_id("user_password")
element.send_keys(password)
element = driver.find_element_by_id("user_password_confirmation")
element.send_keys(password)
element.send_keys(Keys.RETURN)
element.close()
```



Practical Demonstration – Selenium Stealth

```
import sys
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium_stealth import stealth

name = sys.argv[1]
email = sys.argv[2]
password = sys.argv[3]

options = webdriver.ChromeOptions()
options.add_argument("--disable-blink-features=AutomationControlled")
driver = webdriver.Chrome(options=options)
stealth(driver,
        languages=["en-US", "en"],
        vendor="Google Inc.",
        platform="Win32",
        webgl_vendor="Intel Inc.",
        renderer="Intel Iris OpenGL Engine",
        fix_hairline=True,
    )
driver.get("http://demo-rails-app-testing.ipm-corporation.com/signup?disallow_selenium=true")
element = driver.find_element_by_id("user_name")
element.send_keys(name)
element = driver.find_element_by_id("user_email")
element.send_keys(email)
element = driver.find_element_by_id("user_password")
element.send_keys(password)
element = driver.find_element_by_id("user_password_confirmation")
element.send_keys(password)
element.send_keys(Keys.RETURN)
element.close()
```

