RISE OF THE HYPEBOTS
Couldn’t buy anything due to bots

Decided to enter the game in 2012

Unfortunately still in the game.

Looks good tho?
Why Bots?

We Built the Internet For Them.
The Bots We Loved

1988
First Internet Bot spotted on IRC

1994
AOL Releases WebCrawler

1999
IRC’s first Malicious Botnet

2007
First “large scale” HTTP botnet
Our (best) Internet

[SEO] Google is not the only bot anymore.

[HTML] “Readable” and Clean HTML is easy to parse

[APIs] Speedy and Reliable

[Capitalism] Purchasing should be easy.
1. If you’re writing tests, you’re one evil genius moment away from becoming a bot reseller

2. If your site is using unit tests, you’re making sure that your site will be easy to bot.
CAPITALISM ENCOURAGES BAD BEHAVIOR
Why We Bot

- Restricted supply increases demand for a heavily marketed product
- Speed run checkout system creates a problem that requires an optimal solution
- Optimal Solution: Computers using computers.
Why “They” Bot

- Restricted supply increases demand for a heavily marketed product
- Buyers market allows for space for a “grey market” to emerge to resell product.
- Need a vast amount of restricted product for more profit
- Optimal Solution: Many computers using computers
How Bots?

Complex Systems
Simple Answers
Types of Bots

Low Level Bot
- Uses API calls to post directly.
- Takes more initial upfront investment.
- Scale to thousands easily.

Browser Based Bot
- Mimics a user by loading a browser.
- Easier to script, and harder to detect.
- More expensive to scale.
Meet the Team!

Monitor-Bot

Account-Bot

Buy-Bot

Sell-Bot
const puppeteer = require('puppeteer')
let browser
let page

beforeAll(async () => {
  browser = await puppeteer.launch()
  page = await browser.newPage()
})

describe('Target Site Homepage', async () => {
  test('has product we want', async () => {
    await page.setViewport({ width: 1280, height: 800 })
    await page.goto('https://www.targetsite.com', { waitUntil: 'networkidle0' })
    const product = await page.$('#targetProduct')
    expect(product).toBeTruthy()
  })
});
const puppeteer = require('puppeteer')
try {
    async () => {
        const browser = await puppeteer.launch({
            headless: true,
        });
        const page = await browser.newPage()
        const navigationPromise = page.waitForNavigation({ timeout: 60000, waitUntil: 'domcontentloaded' })
        await page.setViewport({ width: 1280, height: 800 })
        await page.goto(targetURL, { timeout: 60000, waitUntil: 'domcontentloaded' })
        await page.type('input#Search', productName)
        await page.keyboard.press(String.fromCharCode(13))
        await navigationPromise
        await page.waitForSelector('div.row.abide > div.product-grid-item')
        await page.click('div.product-grid-item > a.image-pg')
        await navigationPromise
        await page.waitForSelector('form.data-product-form')
        await page.click('button[name = "add"]')
        await page.waitFor(2000)
        await navigationPromise
        await page.evaluate(() => {
            document.querySelector('input.cart-form-checkout.button.primary').click();
        });
        await navigationPromise
        await page.waitForSelector('div.main__content')
        await page.$eval('input#checkout_email', el => el.value = 'test@example.com')
        await page.$eval('input#checkout_shipping_address_first_name', el => el.value = 'Test')
        await page.$eval('input#checkout_shipping_address_last_name', el => el.value = 'Test')
        await page.$eval('input#checkout_shipping_address_address1', el => el.value = '123 Test Street')
        await page.$eval('input#checkout_shipping_address_address2', el => el.value = 'Apt 123')
        await page.$eval('input#checkout_shipping_address_city', el => el.value = 'Test City')
A-B-C. A-ALWAYS, B-BE, C-CHECKING. ALWAYS BE CHECKING!
function findItem(url, productName) {
    console.log('looking up item');
    return new Promise(resolve => {
        const options = {
            url: testURL,
            method: 'get',
            headers: {
                'User-Agent': userAgent
            }
        }

        request(options, (err, res, body) => {
            if (err) {
                return console.log(err);
            }
            parseString(body, function(err, result) {
                if (err) {
                    return console.log(err);
                }
                console.log('connected successfully, looking up items');
                const products = result.urlset.url;
                const product = products[0]['image:image'][0]
                resolve(_.filter(products, { 'image:image': [{ 'image:title': [ testProduct ]}]}) });
            });
        });
    });
}
simple_monitor.avi goes here
Run all those requests through a proxy rotating proxy.

Post the monitor results to a chat

Sell access to that chat for $$$
function fancyMonitor(apiKey) {
    const firstCallValidation = (response) => {
        return typeof response !== 'undefined' && response.statusCode === 200 && response.body.includes("http:"); 
    }
    return asyncWrapper('http://pubproxy.com/api/proxy?api=$apiKey&type=http&country=us&format=txt&https=true&pos=1').then(function (result) {
        console.log("PROXY URL: " + result.proxy);
        sharedProxy = proxy;
        return findItem(testURL, testProduct, proxy, userAgent); //MONITOR THE XML SITEMAP
    }).then(function (result) {
        //SEND A MESSAGE TO DISCORD TO COLLECT $$$
        if (result.length) {
            const hook = new Discord.WebhookClient(webhook__id, webhook__token);
            hook.send('Product Found', result);
        }
    return fancyMonitor(apiKey); //IF IT FAILS, TRY AGAIN
});}
Monitor multiple sites, strike many outlets, not just one.

Purchase “verified” accounts in bulk.

If it can be a variable, it should be.

Cluster deploys increase chances.
The Economist
Making a profit from hypebeasts
IF THERE'S A MARKET THERE'S A PROFIT
Our (own) Economy

- Account Economy - verified accounts come in packs of 500-10,000 run out of China
- Cook Groups - buy friends who always seem to have the information, the right configs, the right stack overflow links, and a monitoring bot.
- Purchasing Scripts - Prettily wrapped request with a fancy GUI
Investment Opportunities!

**Buy-Bot**
- Fun, flirty, and $1500
- Can resell for profit to redditors
- Does it all!
- Dedicated support network
- Fancy UI made with Twitter Bootstrap!

**Monitor-Bot**
- Subscriptions start at $15
- Comes 3 real friends
- Compatible with Buy-Bot
- Dedicated support network
- Stack overflow links provided
RESELLERS DO RESELLING BEST
```javascript
const puppeteer = require('puppeteer')
try {
  async () => {
    const browser = await puppeteer.launch({
      headless: false, // launch headful mode
    });
    const page = await browser.newPage()
    const navigationPromise = page.waitForNavigation({ timeout: 60000, waitUntil: 'domcontentloaded' })
    await page.setViewport({ width: 1280, height: 800 })
    await page.$eval('form.cart', form => form.submit());
    await navigationPromise
    await page.goto('https://www.aiobot.com/checkout/', { timeout: 60000, waitUntil: 'networkidle2' })
    await page.$eval('input#billing_first_name', el => el.value = firstName)
    await page.$eval('input#billing_last_name', el => el.value = lastName)
    await page.$eval('input#billing_email', el => el.value = email)
    await page.click('input#payment_method_paypal')
    await page.waitFor(2000)
    await page.click('input#terms')
  }()
} catch (err) {
  console.error(err)
}
```
Future Bot
And why we need to get
LOL totally rand0m
We Care about (your) money.

- Resellers are not good consumers, often make more profit than the company does.
- Resellers crowding drops means that real consumers stop purchasing.
- Human's perceive “fairness” and if a company isn't fair, then why give them money?
Blacklisting?

- Bot Traffic Mimicks human traffic, so you’ll end up blacklisting consumers.
- Accounts are purchased and dumped regularly.
- Since bots are deployed in clusters and behind proxies, IPs will change each purchase (or each request).
- Blacklisting means creating a useless database.
False Listings confuse bots and consumers

In person sales just mean in person resellers

Lottery based means bots buy lotto tickets instead
Making the Web Unpredictable Again

- Adding entropy into your system does not mean introducing unreliability.
- Measured unpredictability can make many tasks more difficult.
- Let’s encrypt our process, not just our passwords.
THIS AIN'T A SCENE
IT'S AN ARMS RACE