



# Rock'em Sock'em Robots

Bot Swatting Like The Pros



Aaron Bedra  
Principal Engineer, Groupon  
@abedra  
[keybase.io/abedra](http://keybase.io/abedra)

"Well, there's a judge and a subject, and... the judge asks questions and, depending on the subject's answers, determines who he is talking with... what he is talking with, and, um... All you have to do is ask me a question."

-- Alan Turing, The Imitation Game

Asymmetric warfare

The internet is  
powered by robots

YAHOO!



Yandex

Google

Aol.

We employ teams of  
people to help manage  
good robots

But all robots are not  
created equal

```
10.20.253.8 - - [08/Apr/2015:09:17:52 +0000]
"POST /login HTTP/1.1" 200 267 "-" "curl/
7.35.0" "77.77.165.233"
```



```
10.20.253.8 - - [08/Apr/2015:10:20:21 +0000]
"POST /login HTTP/1.1" 200 267 "-" "Mozilla/
5.0 (Windows NT 6.1; WOW64; rv:8.0) Gecko/
20100101 Firefox/8.0" "77.77.165.233"
```

Some robots are more  
trouble than they are  
worth

How much of your  
traffic is bot related?

How much of it should  
be?

Who here does  
testing/tracking?

How bad do these robots  
throw off your tests?

What else are bots  
doing on your site?

Let's talk about  
common types



Spiders

The root of most things  
we will talk about

They are often used  
inside of scrapers and  
scanners to find content

But can be used on  
their own as well

Trivial to build

# How to build a spider

- Go to starting page
- Gather all links on the page and put them into a queue
- Visit link in queue
- Repeat until queue is empty (or sentinel)
- Keep a record of all links visited

Spiders are usually  
easy to detect

They deviate from  
typical behavior quickly



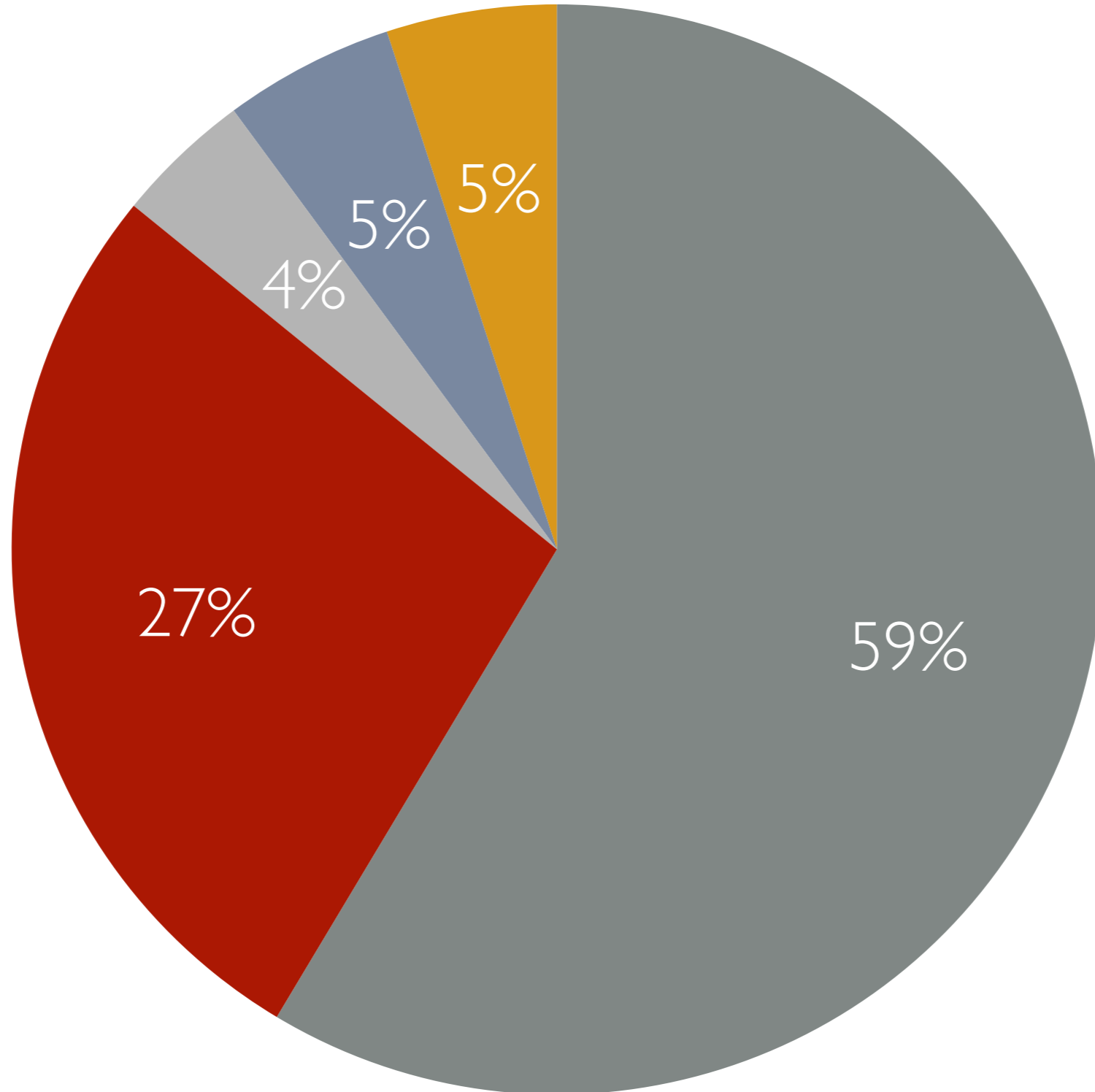
● GET

● POST

● HEAD

● PUT

● DELETE



Simply sampling traffic and  
comparing for deviation  
can usually catch a spider

Velocity can also be  
an indicator

# Scrapers

They want your data

Scenario 1: You  
provide an API

Either stop them outright  
or refer them to the API

Scenario 2: You don't and  
they shouldn't be doing  
this



Stop them

Scenario 3: You don't  
provide an API and you  
should

Stop being lazy

APIs are for machines,  
Web Interfaces are for  
Humans

If there's no reason for a machine, don't allow it\*

Most of the time  
scrapers are dumb

```
<!– <a href="gotcha"></a> –>
```

Start with simple



Accept that a small portion  
of really intelligent scrapers  
will make it through

Detection is similar to  
spiders

In fact, a spider might  
precede a scraper

But behavior deviation is  
still an acceptable  
detection mechanism

# Scanners

Unlike scrapers and spiders, scanners are purely malicious

They are looking for vulnerabilities in your application(s)

They are also pretty  
easy to spot



They deviate from  
normal behavior

They submit obviously  
malicious data

And they produce a  
lot of 404s

You want to block  
these\*

WAFs can help

But prefer running a  
WAF in passive mode

Other

Fraud, (D)DoS,  
Espionage, etc.



Still falls in the  
“malicious” category

But behaves  
differently

Usually has a focused  
target

Almost obviously so

Detection is a little harder  
here, but still follows the  
previous rules

What to look for

# Anomalies

Anything that let's you  
reject  $H_0$



But first you have to  
define “normal”

And what has to change  
to be “not normal”

```
10.20.253.8 - - [08/Apr/2015:08:20:21 +0000]
"POST /login HTTP/1.1" 200 267 "-" "Mozilla/
5.0 (Windows NT 6.1; WOW64; rv:8.0) Gecko/
20100101 Firefox/8.0" "77.77.165.233"
```

```
10.20.253.8 - - [08/Apr/2015:08:20:22 +0000]
"POST /users/king-roland/credit_cards HTTP/
1.1" 302 2085 "-" "Mozilla/5.0 (Windows NT
6.1; WOW64; rv:8.0) Gecko/20100101 Firefox/
8.0" "77.77.165.233"
```

```
10.20.253.8 - - [08/Apr/2015:08:20:23 +0000]  
"POST /users/king-roland/credit_cards HTTP/  
1.1" 302 2083 "-" "Mozilla/5.0 (Windows NT  
6.1; WOW64; rv:8.0) Gecko/20100101 Firefox/  
8.0" "77.77.165.233"
```

```
10.20.253.8 - - [08/Apr/2015:08:20:24 +0000]
"POST /users/king-roland/credit_cards HTTP/
1.1" 302 2085 "-" "Mozilla/5.0 (Windows NT
6.1; WOW64; rv:8.0) Gecko/20100101 Firefox/
8.0" "77.77.165.233"
```

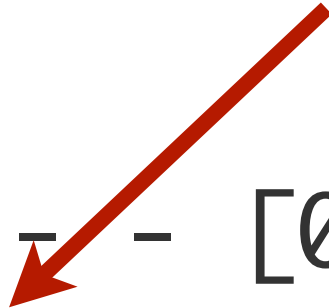
What do you see?

I see a carding attack



!?!?



## Login Request



```
10.20.253.8 - - [08/Apr/2015:08:20:21 +0000]  
"POST /login HTTP/1.1" 200 267 "-" "Mozilla/  
5.0 (Windows NT 6.1; WOW64; rv:8.0) Gecko/  
20100101 Firefox/8.0" "77.77.165.233"
```

Add credit card to account #1

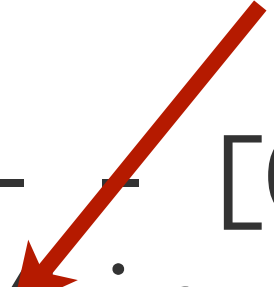
1 sec delay



```
10.20.253.8 - - [08/Apr/2015:08:20:22 +0000]
"POST /users/king-roland/credit_cards HTTP/
1.1" 302 2085 "-" "Mozilla/5.0 (Windows NT
6.1; WOW64; rv:8.0) Gecko/20100101 Firefox/
8.0" "77.77.165.233"
```

## Add credit card to account #2

1 sec delay



```
10.20.253.8 - - [08/Apr/2015:08:20:23 +0000]
"POST /users/king-roland/credit_cards HTTP/
1.1" 302 2083 "-" "Mozilla/5.0 (Windows NT
6.1; WOW64; rv:8.0) Gecko/20100101 Firefox/
8.0" "77.77.165.233"
```



FF 8 on Windows 7  
or Bot?

## Add credit card to account #3

```
10.20.253.8 - - [08/Apr/2015:08:20:24 +0000]  
"POST /users/king-roland/credit_cards HTTP/  
1.1" 302 2085 "-" "Mozilla/5.0 (Windows NT  
6.1; WOW64; rv:8.0) Gecko/20100101 Firefox/  
8.0" "77.77.165.233"
```

Plovdiv Bulgaria

1 sec delay

FF 8 on Windows 7  
or Bot?

## Add credit card to account #3

1 sec delay

```
10.20.253.8 - - [08/Apr/2015:08:20:24 +0000]  
"POST /users/king-roland/credit_cards HTTP/  
1.1" 302 2085 "-" "Mozilla/5.0 (Windows NT  
6.1; WOW64; rv:8.0) Gecko/20100101 Firefox/  
8.0" "77.77.165.233"
```

Doesn't follow 302

FF 8 on Windows 7  
or Bot?

Plovdiv Bulgaria

And this continues

10,000 more times



Behavior deviation

Velocity

Access pattern

Time of day

# Geo Location

HTTP verb distribution

User Agent

Header order



Success rate

	209.0.141.228	AS32479 OPENBAND MULTIMEDIA, LLC	Reston, Virginia, United States	1010	0	0	0	1010	100	0	100	0	0	<a href="#">Details</a>	<a href="#">Whitelist</a>	<a href="#">Blacklist</a>
--	---------------	---	---------------------------------------	------	---	---	---	------	-----	---	-----	---	---	-------------------------	---------------------------	---------------------------

### Excessive Response Codes

IP	Requests	% Success	# 403	%403	# 404	% 404	# 4xx	% 4xx	# 5xx	% 5xx
209.0.141.228	1878	0.00	0	0.00	1878	100.00	1878	100.00	0	0.00

Going deeper

“Of course machines can't think as people do. A machine is different from a person. Hence, they think differently.”

-- Alan Turing, The Imitation Game

What's our goal?

Block robots as  
quickly as possible

Embed detection scripts  
in your applications

They should gather  
information and POST  
back to you



JS can do a lot

developer.mozilla.org/en-  
US/docs/Web/API/  
Navigator

```
var ua = navigator.userAgent;
var resolution = function () {
    var dimensions = (screen.height > screen.width) ?
                    [screen.height, screen.width] :
                    [screen.width, screen.height];
    if (dimensions !== "undefined") {
        return dimensions;
    }
}
var platform = function () {
    if (navigator.platform) {
        return navigator.platform;
    }
}
```

You can also use  
Flash

The details that you  
gather can make it really  
easy to spot a bot

If it doesn't execute it's  
probably a bot\*

But there's a lot to  
examine

User Agent



# Screen Resolution

# Cursor movement pattern

What plugins are  
installed?

Fingerprint(s)

Store the fingerprints  
of known bots

github.com/Valve/  
fingerprintjs

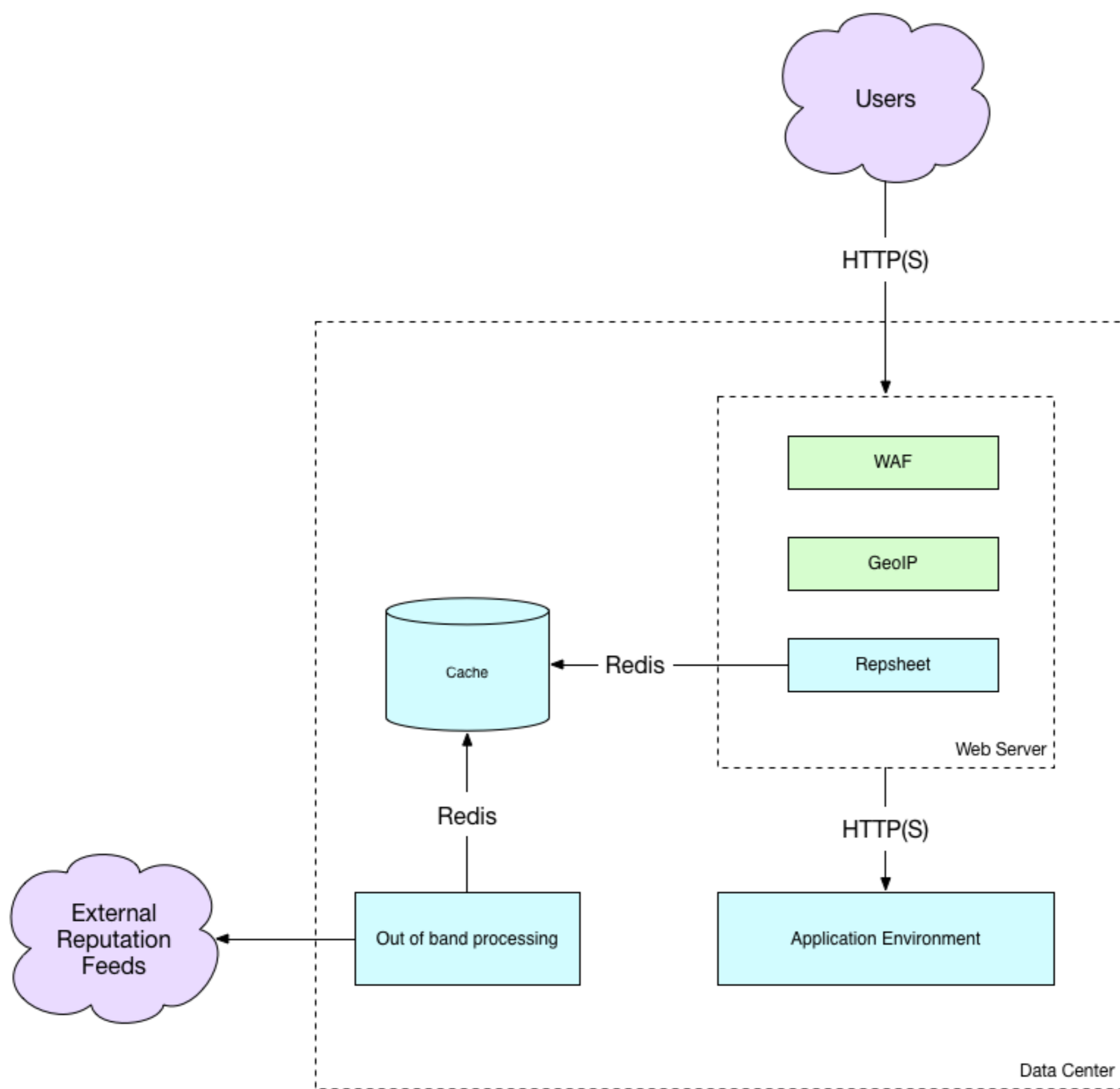
Wrapping up

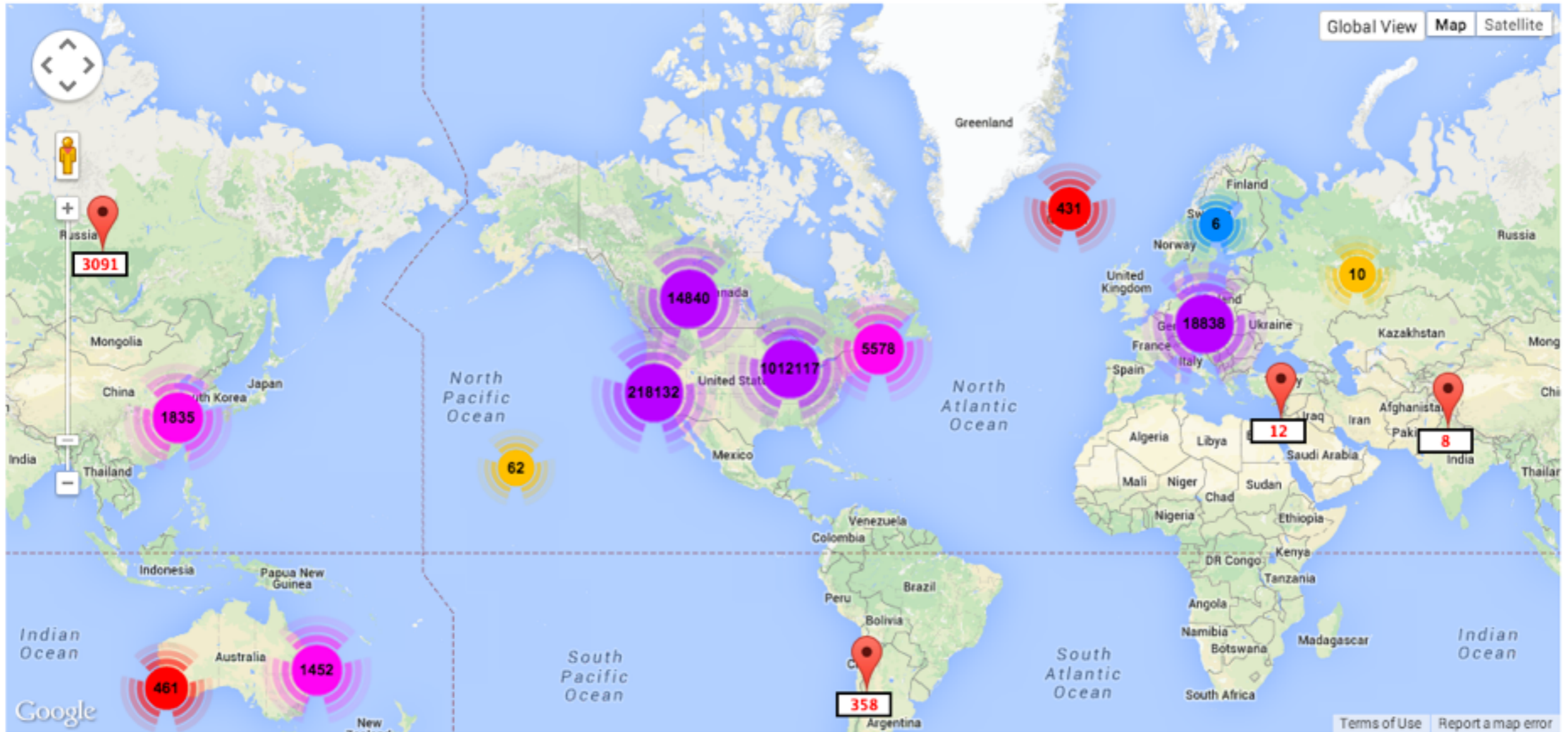
We employ teams of  
people to manage the  
good robots



Maybe it's time to hire a  
team of people that  
manages the bad ones too

We need to build systems  
that do this detection





_entity_	asn	count	location	permalink	reason	violations
<a href="#">64.233.172.32</a>	AS15169 Google Inc.	142597	Mountain View, California, United States		<a href="#">ip-response-code+block</a>	142580
<a href="#">69.47.170.204</a>	AS12083 WideOpenWest Finance LLC	71894	Glen Ellyn, Illinois, United States		<a href="#">ip-blocked-history+block ip-response-code+block</a>	1587456
<a href="#">23.23.189.197</a>	AS14618 Amazon.com, Inc.	44812	Ashburn, Virginia, United States		<a href="#">ip-blocked-history+block ip-response-code+block</a>	751546

Reduce the noise

Reduce the impact of  
attacks

Improve confidence in  
your data

# Questions?

- [github.com/repsheet](https://github.com/repsheet)
- [developer.mozilla.org/en-US/docs/Web/API/Navigator](https://developer.mozilla.org/en-US/docs/Web/API/Navigator)
- [github.com/Valve/fingerprintjs](https://github.com/Valve/fingerprintjs)
- [github.com/Valve/fingerprintjs2](https://github.com/Valve/fingerprintjs2)