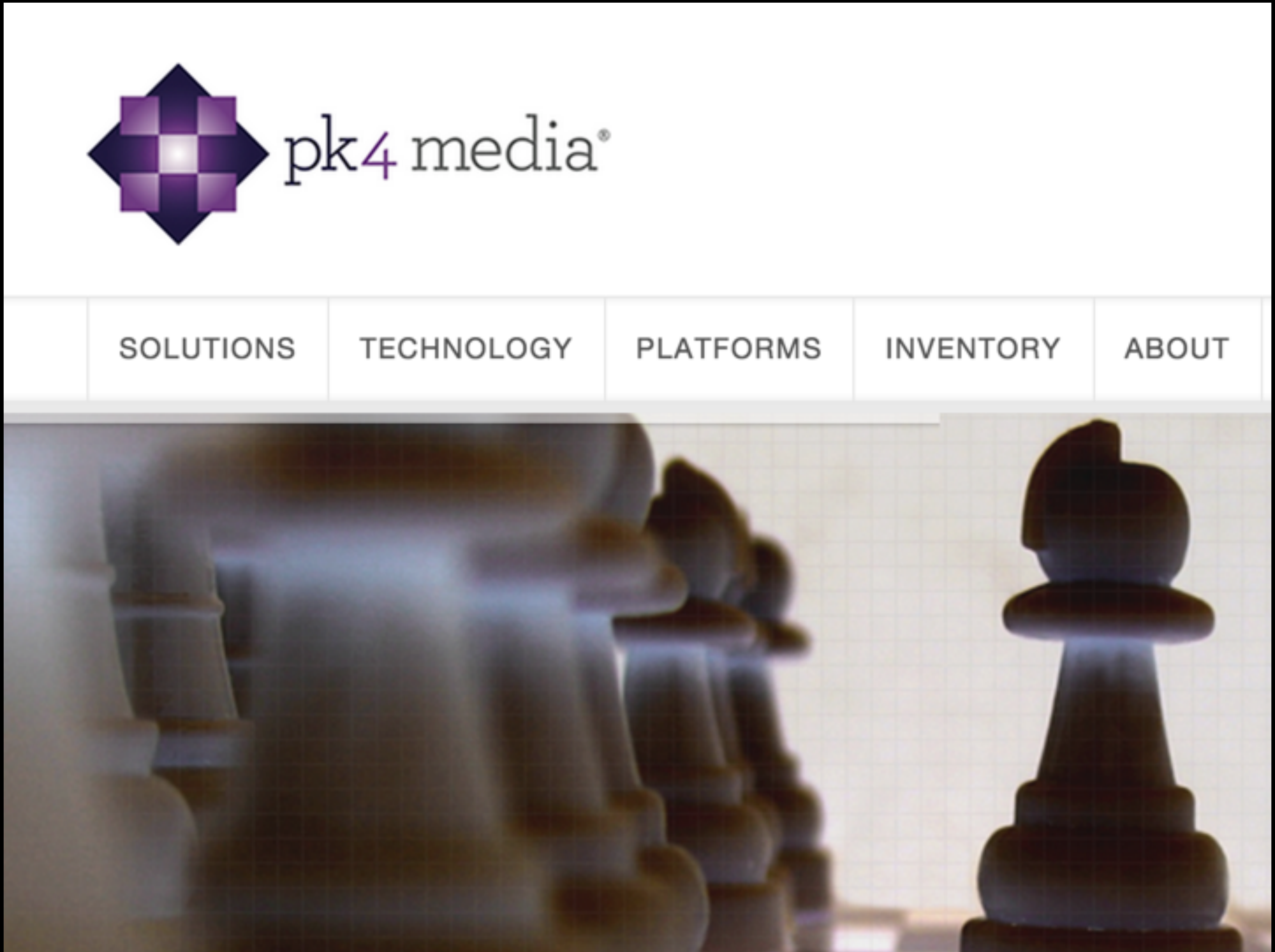
making interactive maps in d3

la-front-end

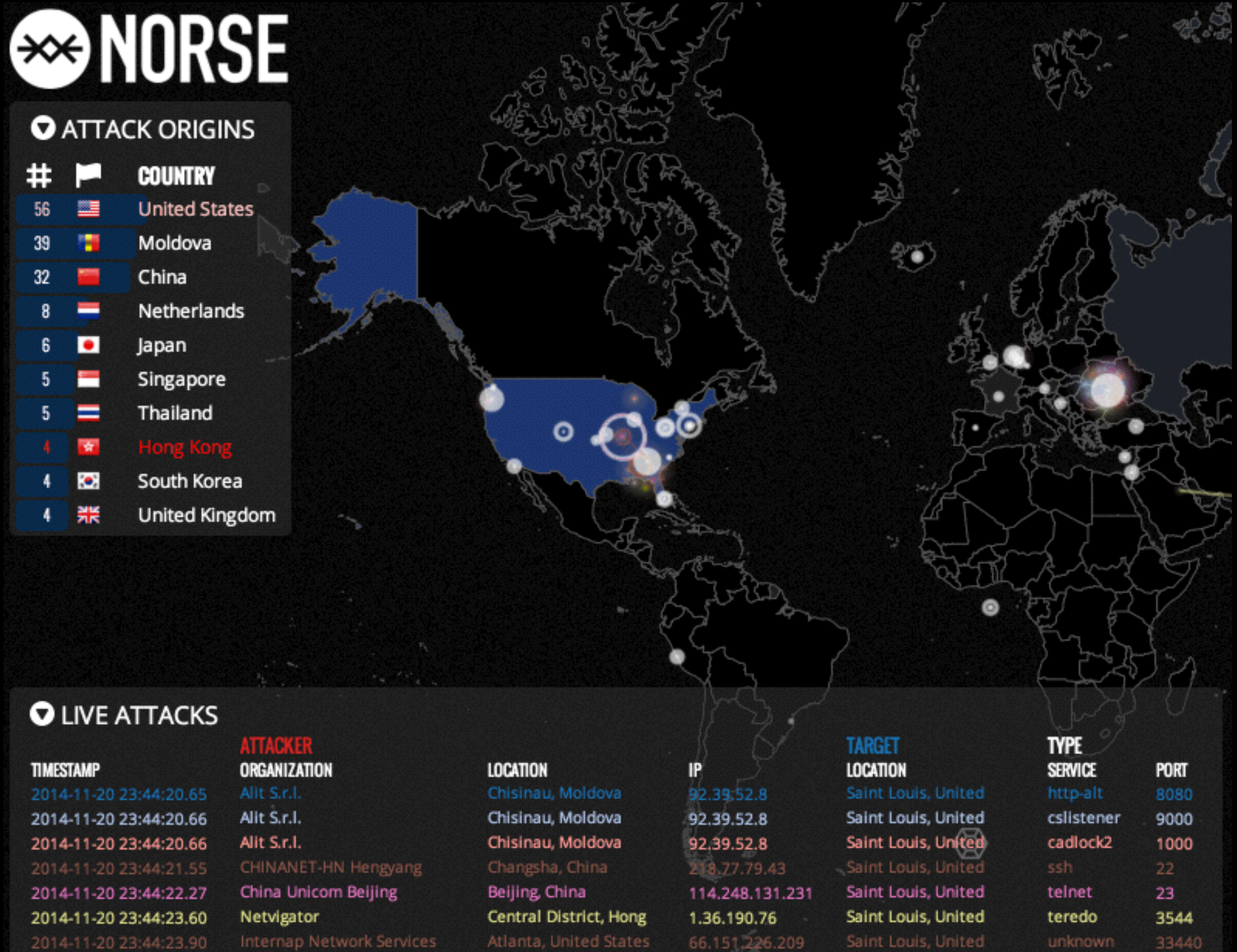
github.com/standyro/d3-usa-map

idea sparked by

redesign of my company's corporate website



- ✓ real time behavior
- ✓ high interactivity

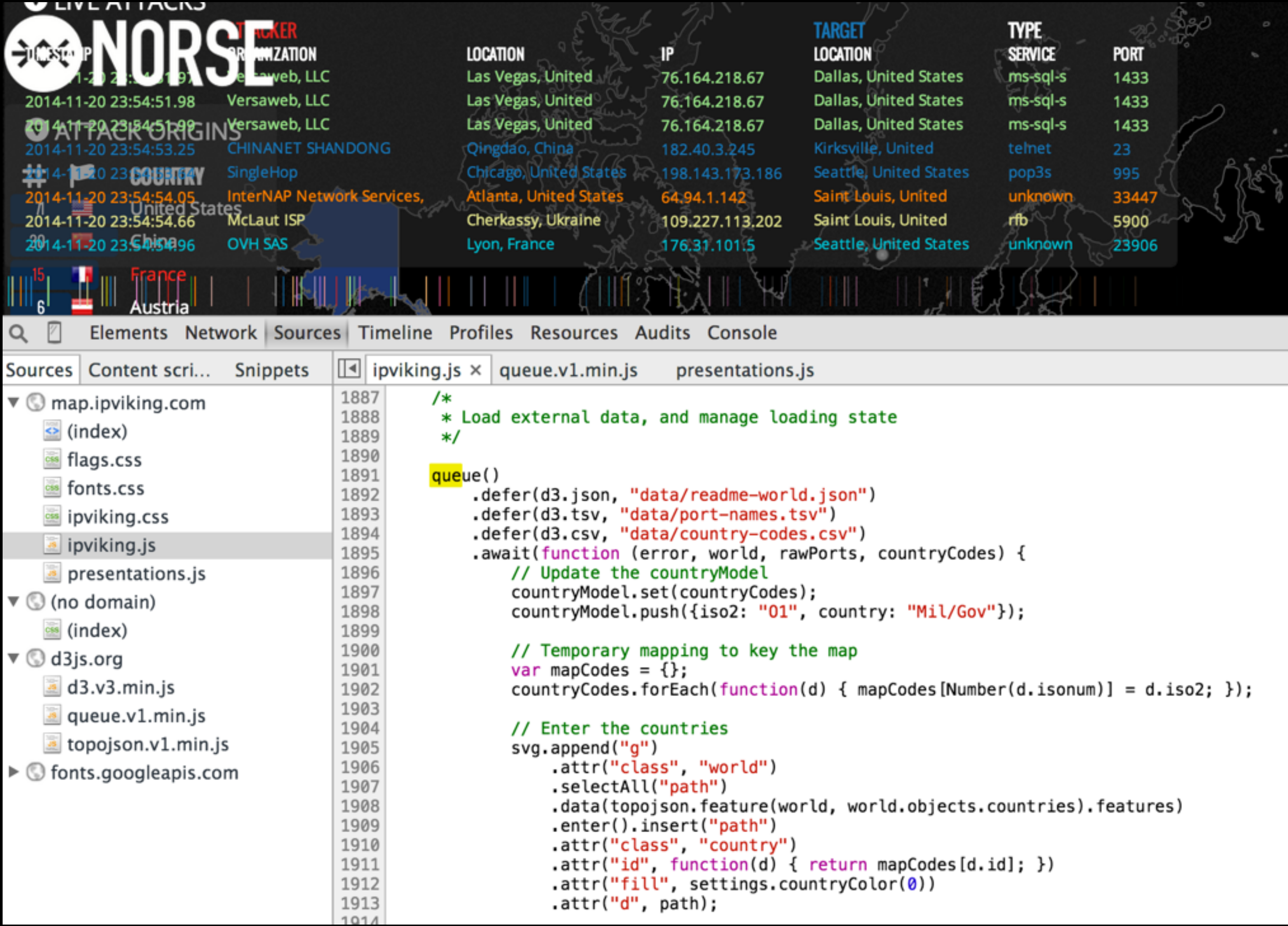


ATTACK ORIGINS

#		COUNTRY
56		United States
39		Moldova
32		China
8		Netherlands
6		Japan
5		Singapore
5		Thailand
4		Hong Kong
4		South Korea
4		United Kingdom

LIVE ATTACKS

	ATTACKER			TARGET	TYPE	
TIMESTAMP	ORGANIZATION	LOCATION	IP	LOCATION	SERVICE	PORT
2014-11-20 23:44:20.65	Alit S.r.l.	Chisinau, Moldova	92.39.52.8	Saint Louis, United	http-alt	8080
2014-11-20 23:44:20.66	Alit S.r.l.	Chisinau, Moldova	92.39.52.8	Saint Louis, United	cslistener	9000
2014-11-20 23:44:20.66	Alit S.r.l.	Chisinau, Moldova	92.39.52.8	Saint Louis, United	cadlock2	1000
2014-11-20 23:44:21.55	CHINANET-HN Hengyang	Changsha, China	218.77.79.43	Saint Louis, United	ssh	22
2014-11-20 23:44:22.27	China Unicom Beijing	Beijing, China	114.248.131.231	Saint Louis, United	telnet	23
2014-11-20 23:44:23.60	Netvigator	Central District, Hong	1.36.190.76	Saint Louis, United	teredo	3544
2014-11-20 23:44:23.90	Internap Network Services	Atlanta, United States	66.151.226.209	Saint Louis, United	unknown	33440



TIMESTAMP	ATTACKER ORGANIZATION	LOCATION	IP	TARGET LOCATION	TYPE SERVICE	PORT
2014-11-20 23:54:51.97	Versaweb, LLC	Las Vegas, United	76.164.218.67	Dallas, United States	ms-sql-s	1433
2014-11-20 23:54:51.98	Versaweb, LLC	Las Vegas, United	76.164.218.67	Dallas, United States	ms-sql-s	1433
2014-11-20 23:54:51.99	Versaweb, LLC	Las Vegas, United	76.164.218.67	Dallas, United States	ms-sql-s	1433
2014-11-20 23:54:53.25	CHINANET SHANDONG	Qingdao, China	182.40.3.245	Kirksville, United	telnet	23
2014-11-20 23:54:53.64	SingleHop	Chicago, United States	198.143.173.186	Seattle, United States	pop3s	995
2014-11-20 23:54:54.05	InterNAP Network Services,	Atlanta, United States	64.94.1.142	Saint Louis, United	unknown	33447
2014-11-20 23:54:54.66	McLaut ISP	Cherkassy, Ukraine	109.227.113.202	Saint Louis, United	rfb	5900
2014-11-20 23:54:54.96	OVH SAS	Lyon, France	176.31.101.5	Seattle, United States	unknown	23906

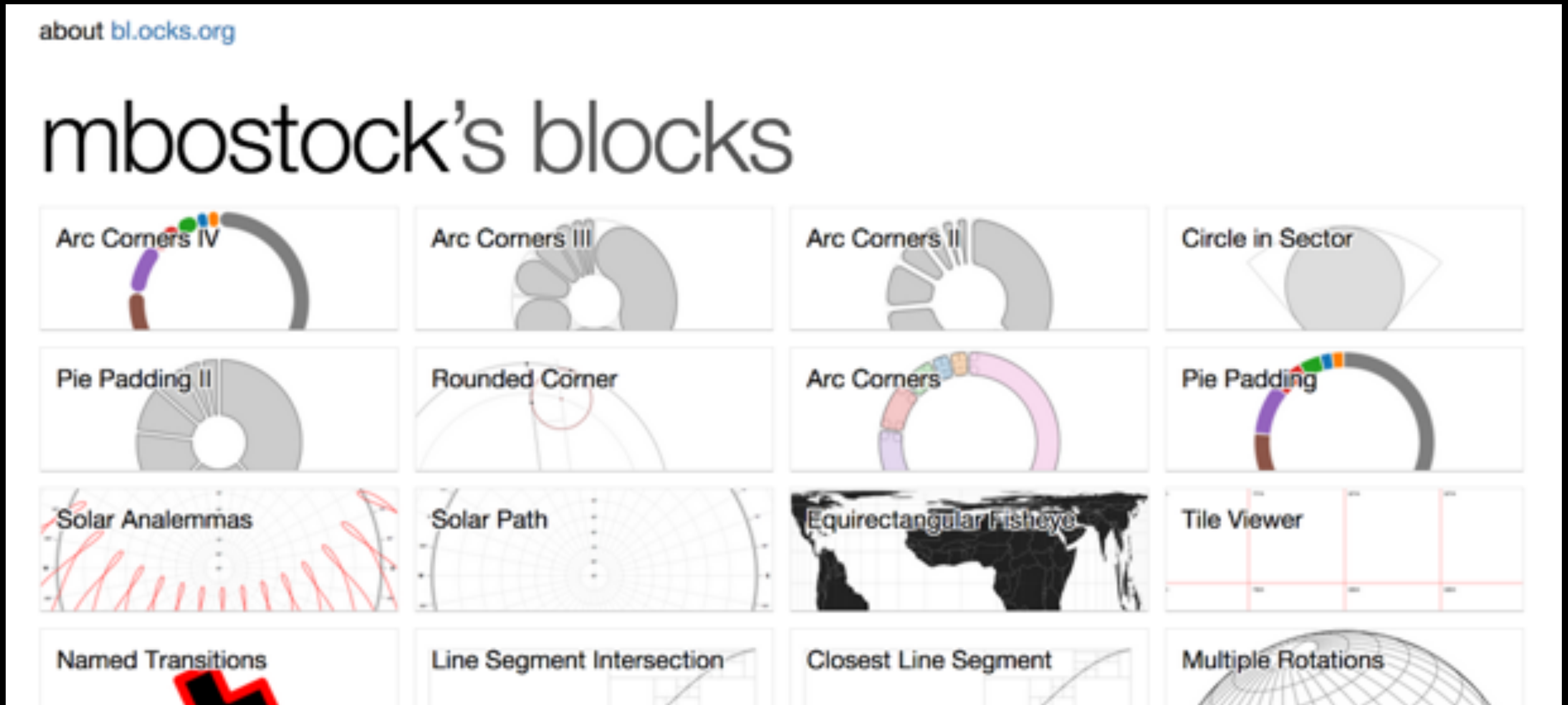
```javascript
/*
 * Load external data, and manage loading state
 */

queue()
    .defer(d3.json, "data/readme-world.json")
    .defer(d3.tsv, "data/port-names.tsv")
    .defer(d3.csv, "data/country-codes.csv")
    .await(function (error, world, rawPorts, countryCodes) {
        // Update the countryModel
        countryModel.set(countryCodes);
        countryModel.push({iso2: "O1", country: "Mil/Gov"});

        // Temporary mapping to key the map
        var mapCodes = {};
        countryCodes.forEach(function(d) { mapCodes[Number(d.isonum)] = d.iso2; });

        // Enter the countries
        svg.append("g")
            .attr("class", "world")
            .selectAll("path")
            .data(topojson.feature(world, world.objects.countries).features)
            .enter().insert("path")
            .attr("class", "country")
            .attr("id", function(d) { return mapCodes[d.id]; })
            .attr("fill", settings.countryColor(0))
            .attr("d", path);
```

this looks familiar!

as with most things d3 related
the Norse projection is heavily
influenced by mike bostock's work
with some websocket work around it

http://bl.ocks.org/mbostock

check
out ->



about bl.ocks.org

mbostock's blocks

norse uses country codes for objects
within map

our business is primarily
oriented around US customers
and advertising centric

so instead of countries
use city regions / nielsen DMA
                (designated market area)
                    TV marketing term

show where ads are
being delivered in real time

in each region,
possibly
simultaneous

d3 map projection

topojson

https://github.com/mbostock/topojson

smaller file
sizes than geojson

```
{
  "type": "Topology",
  "transform": {
    "scale": [
      0.0057789429942994З,
      0.0024842606260626О7
    ],
    "translate": [
      -124.732975,
      24.544237
    ]
  },
  "objects": {
    "nielsen_dma": {
      "type": "GeometryCollection",
      "geometries": [
        {
          "type": "Polygon",
          "arcs": [
            [0, 1, 2, 3, 4, 5, 6, 7]
          ],
          "id": 662,
          "properties": {
            "name": "dma:",
            "latitude": 32.404348,
            "tvperc": 89.2,
            "dma": 662,
            "dma1": "Abilene-Sweetwater, TX",
            "cableperc": 38.2,
            "adsperc": 51.8,
var width = 960;
var height = 500;

var projection = d3.geo.albers()
  .scale(1070)
  .translate([width / 2, height / 2]);

var path = d3.geo.path().projection(projection);

var svg = d3.select(“body")
  .append("svg")
  .attr("width", 640)
  .attr("height", 350)
  .attr("background-color", '#ccc')
```

```
queue()
  .defer(d3.json, "data/dma.json")
  .defer(d3.csv, "data/dma.csv")
  .await(function (error, dmaMap, dmaData) {
    svg.append(“g")
      .attr("class", "world")
      .selectAll("path")
      .data(
        topojson
        .feature(dmaMap, dmaMap.objects.nielsen_dma)
        .features
      )
      .enter()
      .append("path")
      .attr("class", "dma")
      .attr("id", function(d) { return d.id; })
      .attr("d", path);
```

```
var socket = io();

. . .

    socket.on('event', function (data) {
      console.dir('SOCKET EVENT');
      var res = data.split(':');

      var dmaCode = res[0];
      var dmaName = res[1];
      var device = res[2];
      var browser = res[3];

      console.dir(device);

      $('#' + dmaCode).attr("class", "dma-highlight");
      setTimeout(function() {
        $('#' + dmaCode).attr("class", "dma");
      }, 1000);
```

```
var $tbody = $('#events').find('tbody');

if ($tbody.children('tr').length > 5) {
  $tbody.children().last().remove();
}

$tbody.prepend($('<tr>')
    .append($('<td>')
      .append(dmaName)
    )
    .append($('<td>')
      .append(dmaCode)
    )
    .append($('<td>')
     .append(device)
    )
    .append($('<td>')
      .append(browser)
    )
  );
```

```css
#map {
  margin: 20px;
}

#map svg {
  display: block;
  margin: auto;
}

path {
  -webkit-transition: fill 0.5s ease-out;
  transition: fill 0.5s ease-out;
  stroke: #4DB6AC;
  stroke-width: 0.5;
}

.dma { fill: #009688; }
.dma-highlight { fill: #B2DFDB; }
.dma:hover { fill: #B2DFDB; }
```

server

node JS™

+

SOCKET IO

```
var express = require('express');
var app = express();
var http = require('http').Server(app);
var io = require('socket.io')(http);
var fs = require('fs');
var parse = require('csv-parse');
var _ = require('lodash');

app.use(express.static(__dirname + '/public'))

app.get('/', function(req, res){
  res.sendfile('index.html');
});
```

```
var dmas = [];
var devices = ['mobile', 'tablet', 'pc'];
var browsers = ['IE 9', 'IE 10', 'Firefox 21’,
                'Safari 6', 'WebKit', 'iOS 7', 'iOS 8'];

fs.readFile('public/data/dma.csv', 'utf8', function(err, data) {
  parse(data, {comment: '#'}, function(err, output) {
    output.forEach(function(dma, i) {
      dmas[i] = {
        'id': dma[1],
        'name': dma[0]
      };
    })
  })
})
```

```
io.on('connection', function(socket){
  setInterval(function() {
    var randomDma = dmas[_.random(0, dmas.length - 1)];
    var dmaId = null;
    var dmaName = null;
    var device = devices[_.random(0, devices.length - 1)]
    var browser = browsers[_.random(0, browsers.length - 1)]

    if (randomDma && randomDma.id) {
      dmaId = randomDma['id']
      dmaName = randomDma['name'];
    }

    var event = dmaId+':'+dmaName+':'+device+':'+browser;

    io.emit('event', event);
  }, 500);
});

http.listen(3000, function(){
  console.log('listening on *:3000');
});
```

final result

presentation tip

```
brew install highlight
```

```
highlight -O rtf -t 2 -K 40 -k 'Font'
--style example_theme
file_to_be_highlighted.js | pbcopy
```