Mapping “Badness”

“Even before you understand them, your brain is drawn to maps”

Ken Jennings, author and Jeopardy champ

We have looked Maps have always been in the realm of cartographers and artists, but the barrier to entry realm has been steadily opening over the last few decades. Technology computers and the Internet , now any yahoo can map out data on Google (that play on words was odd and intentional). And as we saw in Chapter 4, it’s possible to pull geo-location of IP addresses and get spatial data, and that’s what we’re going to focus on.

How do maps help us within information security?

The data we are going to use for the first half of this chapter comes to us from Symantec, who shared a list of clients infected with the ZeroAccess Botnet as collected over a 24 hour period. But they didn’t share the IP But rather than share the IP address

The question we’ll really want to target in this chapter though, is whether or not we should map the virtual world into the physical?

24 hour period

Who is more likely to get a bot infection?

Simplifying Maps

It’s easy to get all wrapped up thinking that visualizing spatial data (maps) are special, complicated or will somehow take a lot more effort. But with the right tools (and there are plenty available) working with spatial data can not only be relatively simple, but pretty darn fun. In order take some of the mystique out of maps, let’s start by loading up the latitude and longitude points we got from Symantec and just treat them as x,y coordinates and create a simple scatter plot:

# read the CSV with headers

**za <- read.csv("ch5/data/zeroaccess.csv", header=F)**

# create a scatter plot ggplot instance

**gg <- ggplot(data=za, aes(x=long, y=lat))**

# now just add the points, set transparency to 1/20th

**gg <- gg + geom\_point(size=1, color="#000099", alpha=1/20)**

# add axes labels

**gg <- gg + xlab("Longitude") + ylab("Latitude")**

# simplify the theme for aesthetics

**gg <- gg + theme\_bw()**

**print(gg)**

Figure 5.1 Basic Scatterplot using Latitude and Longitude [FILENAME 793725c05f001]

See the map created out of points in figure 5.1? This works with our data because we have over 800,000 data and one point is covering more than a large city. We made it a little less menacing by setting the alpha (transparency of the color as we covered in chapter 3) to be 1/40th of a full color. But just from this basic scatter plot, we can see the density in the eastern half and west coast of the U.S. and most of Europe is covered. We see some concentration around São Paulo, Brazil in South America, and India is outlined quite well. One interesting thing to note here is that China has almost no density and Japan is clearly visible. At this point, we can only make guesses as to what’s going on, but we will end up digging a bit deeper here.

Now comes the secret sauce about maps: it’s all about the projection. Everyone is familiar with the Cartesian coordinate system, if not by name then by sight. It’s the fancy name for coordinate system of the simple x,y plotting we just did. By plotting the latitude and longitude points as same-spaced x and y coordinates, we can see the map, but it looks a little odd. South America is really long for example. And this is where map projections come in because we are projecting three-dimensional spherical world onto a two-dimensional canvas. This creates some problems since there are multiple ways to do that projection as we see in figure 5.3

But creating points on the canvas isn’t all that helpful, we want to have a full canvas to populate. Luckily, within R we’ve got all that data to populate and we can load up map data with a single command:

# load map data of the world

**world <- map\_data(“world”)**

This loads up just over 25 thousand rows describing various things about the world map (less after we filter out Antartica). We can view countries and borders of the world map by tracing a path on the lat/long pairs in the map data. We have to group the paths by the column labeled group (which in this data groups the country). We will then call coord\_map() to create the map projections and we’ll use a simple black and white theme on it.

# load map data of the world

**world <- map\_data(“world”)**

Using a simple Mercator projection, we can load up data for a world map and trace around the country are taking a spherical object

may pointsIt’s easy to get all wrapped up thinking that maps are somehow special, complicated or will somehow take a lot more effort. But we will start very simple and we’ll discover it’s but we’ll walk through them and show how simple they can be. For example, let’s begin by loading up the latitude and longitude points we got from Symantec and just treat them as x,y coordinates and create a simple scatter plot:

Chapter Use Case: ZeroAccess Botnet Analysis

What can you learn from just a set of lat/lon pairs?

Getting basic lat/long metadata (lat/lon -> country/city)

Visualizing lat/long data

Choropleth

Dot plot

Getting more advanced metadata (internet user population & income)

The quest for correlation

WEB CONTENT: ZeroAccess code & visuals

points on a globe

points in the U.S.

setting alpha on the map points

What is the bot infection per state given the # of internet users?

(choropleth on a state basis)