Security Analysis on Blocked Wireless and Wired Network Clients

Jonathan Wilson and JC Thomas

November 27, 2019

Introduction

Cyber security analysts at BYU are interested in which network client type (wireless or wired) is at higher risk for security vulnerabilities. Clients who are attempting to access sites that are listed as blocked by the BYU security are in a higher risk category than clients who do not hit those sites. Clients attempting to visit these sites are blocked. The main motivation for understanding the behavior of wireless versus wired clients is to tighten restrictions on the client type that is at a higher risk. This could better mitigate possible security vulnerabilities that occur in the future.

A wireless client type is defined as any user or machine that is logged onto the BYU network via a wireless connection. A wireless connection is BYU wifi (guests and students) or BYU secure. A wired client type is any user or machine that is logged onto the BYU network via a wired connection such as ethernet. Computers in the libraries, labs, and any computer on campus that uses a wall jack and ethernet cable are examples of wired computers. Sites considered risky (such as possibly containing malware) or inappropriate are blocked by the BYU security team.

The following analysis seeks to find out: Given that a network client is denied access, which client type is denied access more on average?

Through this analysis it was found that wireless clients were at a higher risk classification.

The parameter in question is the average amount of times a client is denied depending on client type.

Methods

The data was obtained by writing an API (application programming interface) that queries from Elasticsearch. Elasticsearch is what BYU security and Network Engineering uses to gather insights about log data, or data about how the network is being used.

We selected fields:

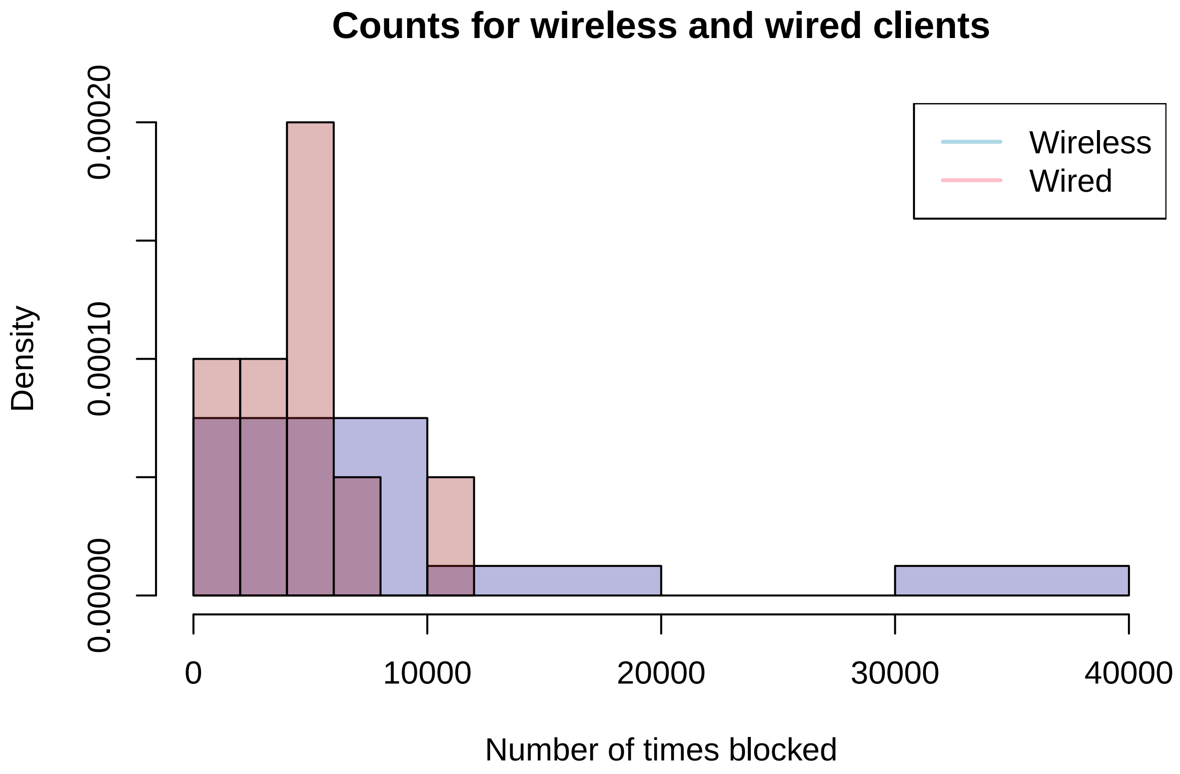
- IP - within an IP range to get wired and wireless clients. Reported as categorical data.
- Blocked - whether or not an IP was blocked. This is a count.

Using Python the data was aggregated within the functionality in the API call and returned a JSON formatted response with IP and Counts as fields. The details of the query will be left out. With this API, IPs and their counts over a determined time span were obtained. The time span was chosen to be three days as the API times out after any longer since the query is going through terabytes of data.

The data gathered from the API was discrete. A summary of the data is given below.

Table 1: Summary of the data

Mean Wireless	Var. Wireless	Mean Wired	Var. Wired
9164.375	97323782	4676.9	8369524



It is observed that for each of the populations the variance of the data is much larger than the mean. It is also seen that this data is right skewed. Considering the skew, discrete data type, mean and variance relationship, and value positivity, a negative binomial distribution was chosen to model the data. The following model is used for both the wireless and wired populations. The model is parameterized in the following way:

$$X_i|r, \theta \sim^{iid} NB(r, \theta)$$

X_i is the count of rejected requests of a certian IP. The parameter r is the size parameter and θ is the probability parameter. Likelihood of X is given as

$$f(data|r, \theta) = \Pi_{i=1}^{n} f(x_i|r, \theta)$$

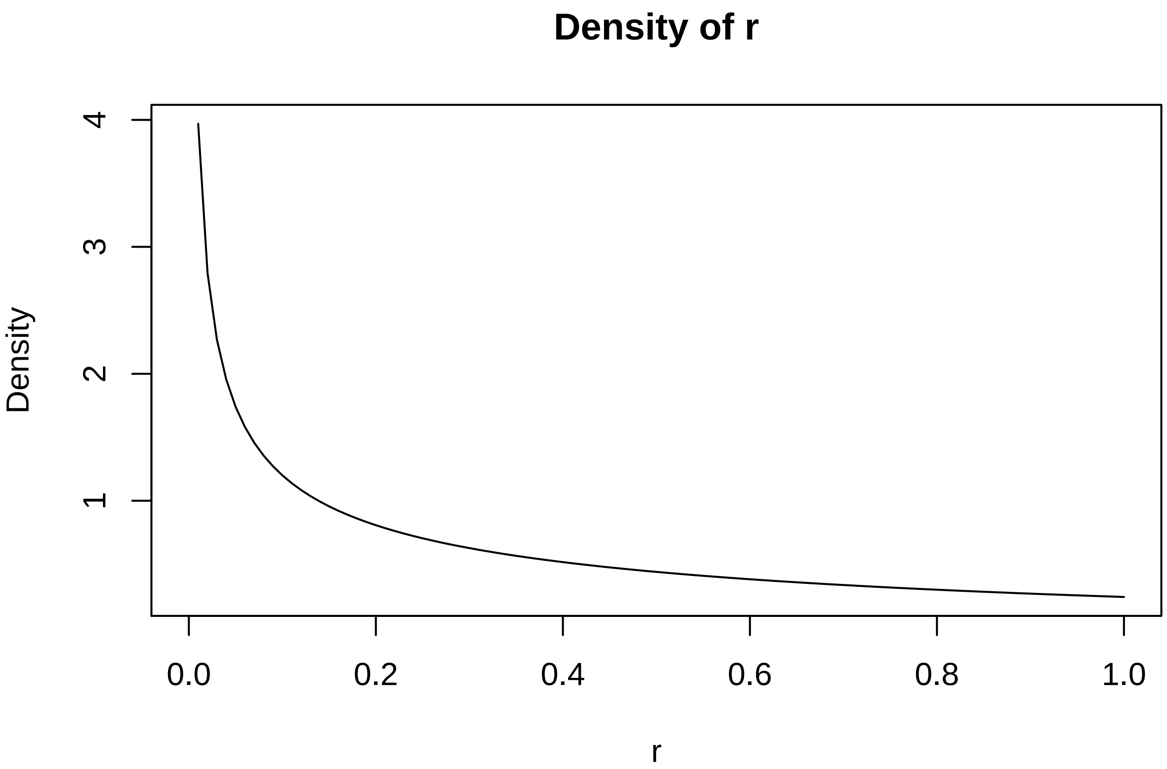
where each x_i follows the negative binomial distribution.

Prior distributions for the parameters r and θ were then chosen. The prior values given were selected for both the wireless and wired populations.

The gamma distribution was chosen to be the prior for r. The gamma distribution is an appropriate choice for representing r due to its continuous and right skewed nature. Since r represents the shape parameter in the negative binomial distribution and the network data is right skewed, it can be expected to observe a shape parameter with high probabilities associated with lower values and smaller, but still likely, values in the upper tail. Generic parameter values were chosen for this distribution for each population due to a lack of prior knowledge about the topic. The parameterization is as follows:

$$r \sim Gamma(\gamma = .5, \phi = .5)$$

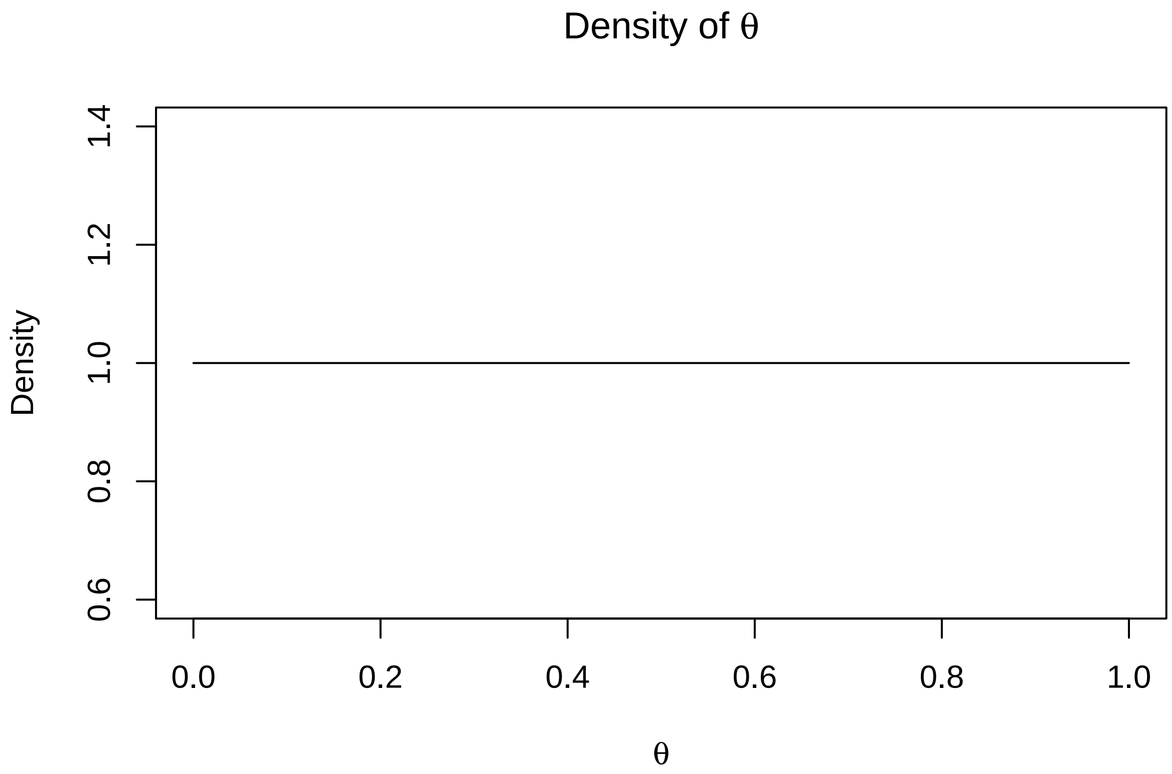
A graph of the density of r is given below.



The beta distribution was selected to be the prior distribution for θ. θ is the probability parameter for the negative binomial distribution. Thus, θ is constrained to be on the interval $[0, 1]$ and must be continuous. The beta distribution fits this description. It is often used as a model for percentages and is a conjugate prior to the negative binomial distribution, making it an excellent prior distribution choice for *theta*. The parameterization is given as follows:

$$\theta \sim Beta(\alpha = 1, \beta = 1)$$

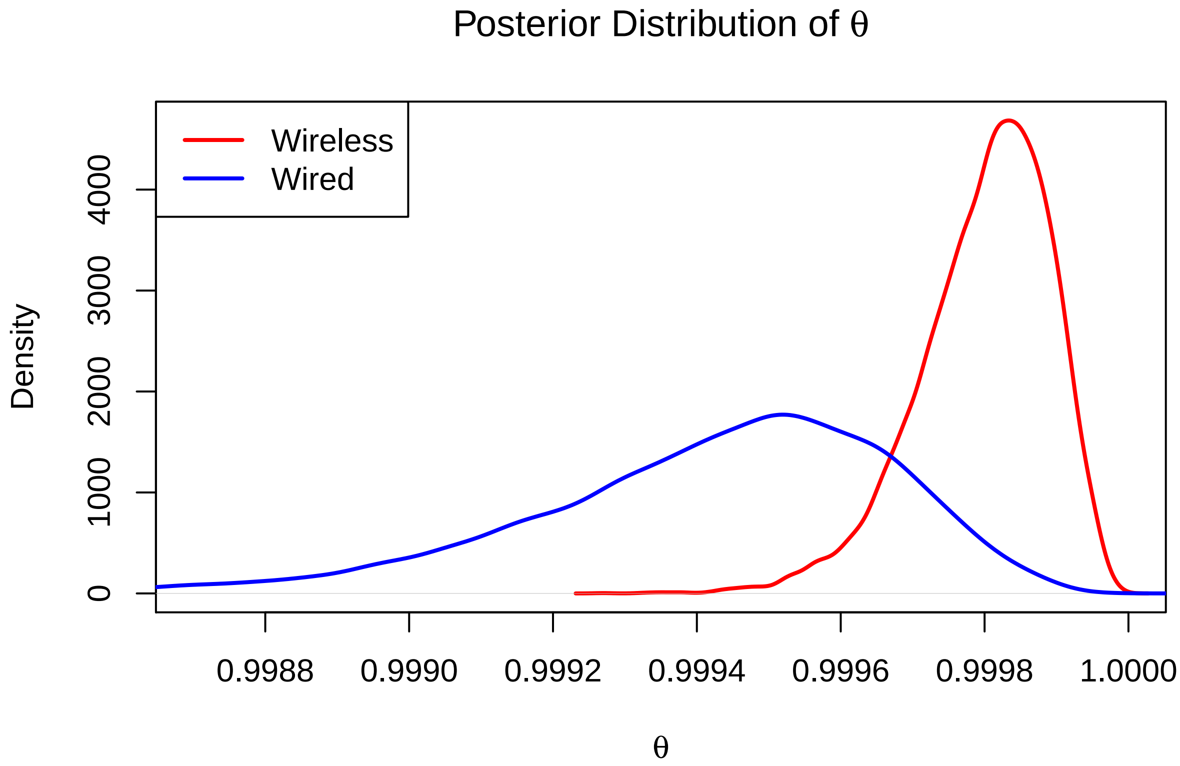
Generic parameter values were chosen for the prior distributions of θ as well. A density plot is given below.



Results

Using a Gibbs sampler with Monte Carlo simulation for the values of θ and a Metropolis Random Walk for the values of r, posterior distributions were produced for θ and r. The posterior distributions and summary statistics for the two parameters for each client type are given below.

Distributions:



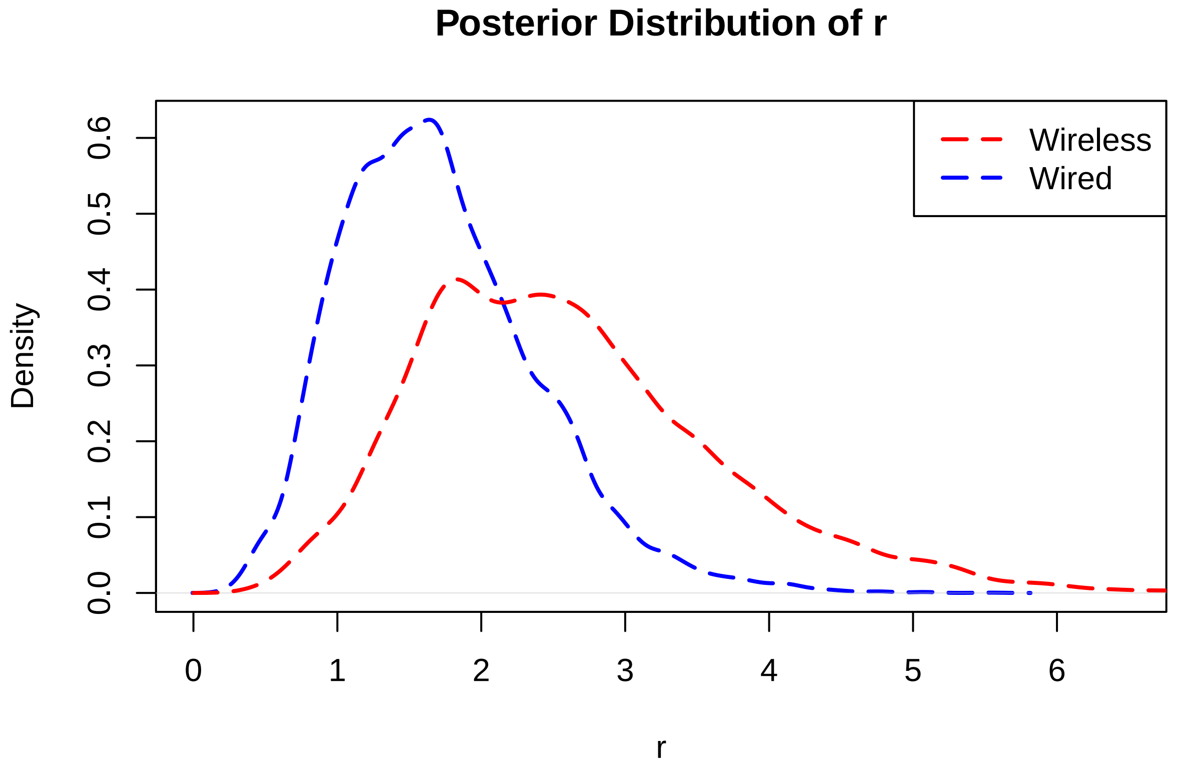


Table 2: Summary Statistics for Posteriors

	95% CrI low	95% CrI Up	Mean	SD
Wireless Theta	0.9996	0.9999	0.9998	0.0001
Wireless r	0.6719	3.2719	1.7129	0.6778
Wired Theta	0.9988	0.9998	0.9994	0.0003
Wired r	0.9551	5.1309	2.6094	1.0556

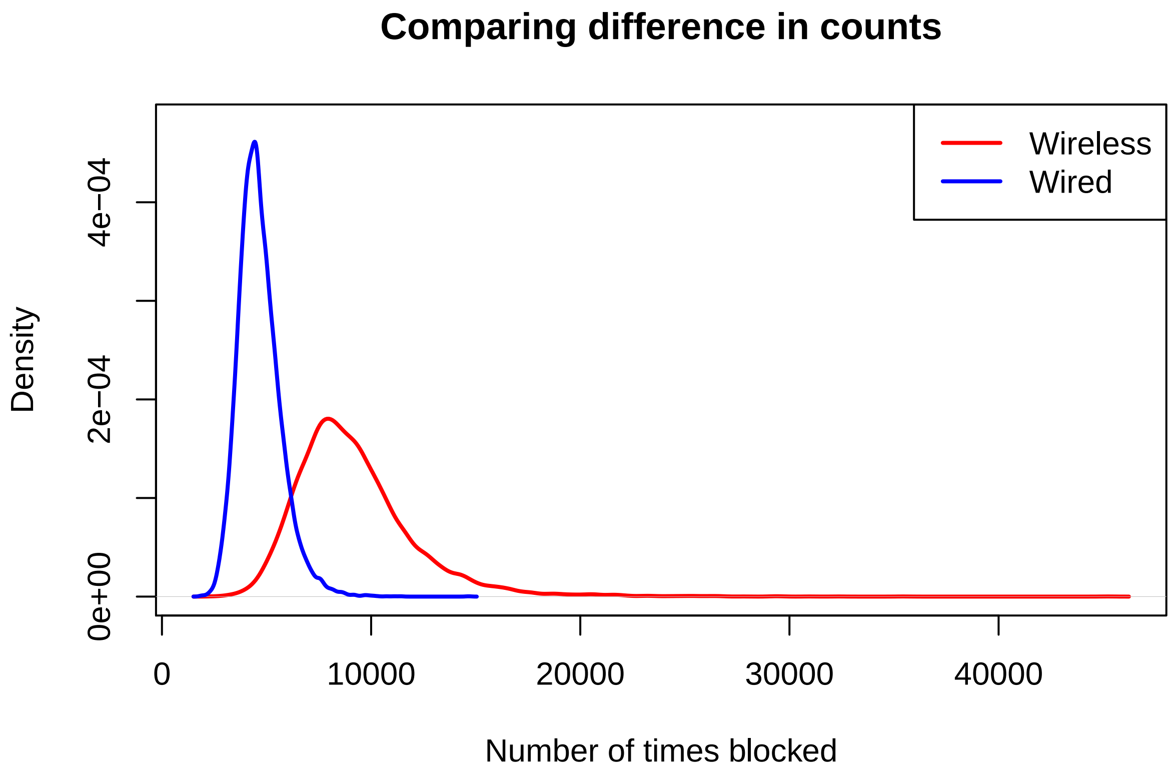
Table 3: Summary of Results

	95% CrI low	95% CrI Up	Mean	SD
Wireless	5110.149	16029.020	9186.266	2862.915
Wired	3045.800	7001.165	4669.260	1007.216

As can be seen, the values of θ for each population are close to 1 and very similar. The θ values for wired clients has a larger spread. The r value for wireless clients has a smaller mean and standard deviation than when compared to the values of r for wired clients.

Conclusions

The parameter of interest is the average count of times denied (given a client is denied). This is calculated for both wireless and wired clients below. Plots and summary statistics are calculated.



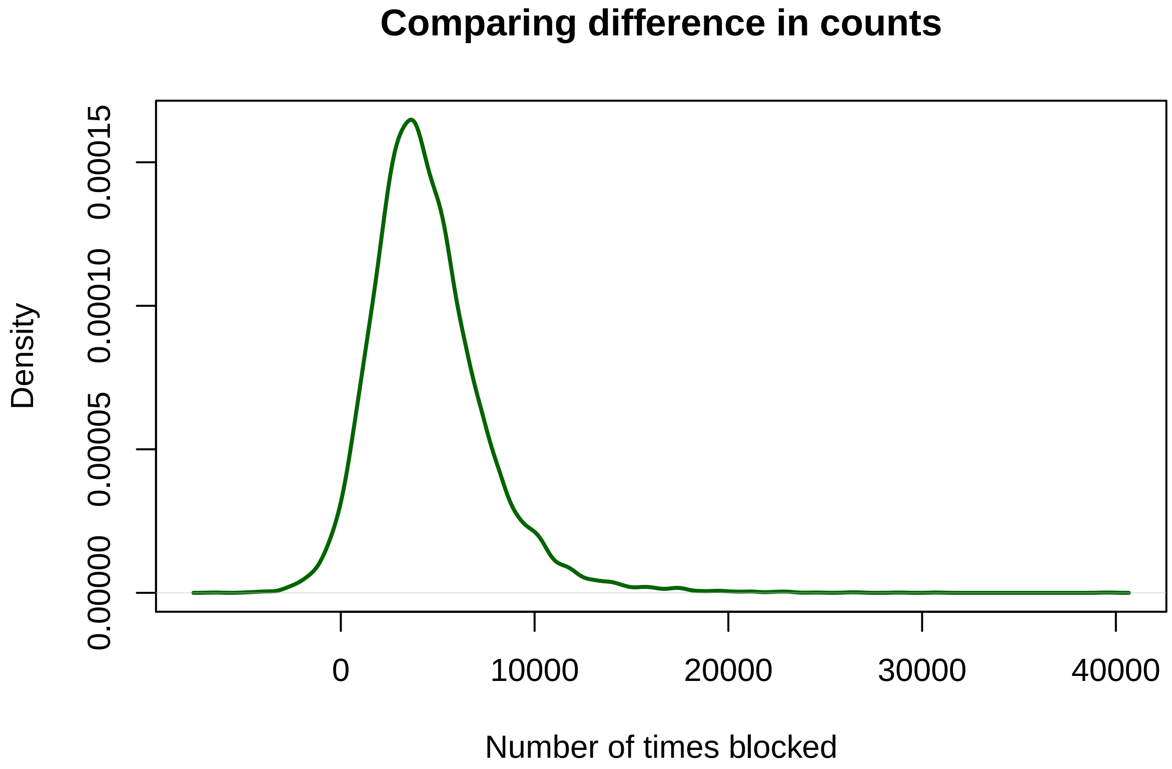
The mean value of the posterior average of times denied is 9186.3 and 4669.3 for wireless and wired clients respectively. The mean average value for wireless clients is about twice as large as it is for wired clients.

Table 4: Summary Statistics on the Differences

90% CrI Low.	90% CrI Up.	Mean Diff.	SD of Diff.	Prob Wireless > Wired
536.539	9870.015	4517.005	3026.146	0.9708

The standard deviation follows a similar trend. We see that the standard deviation of the posterior averages for wireless clients (2862.9) is about 3 times as large as the standard deviation of the posterior averages for wired clients (1007.2). There is a 95% chance that the true value of the average falls in the interval $(5110.1, 16029.0)$ for wireless clients and $(3045.8, 7001.2)$ for wired clients. There is minimal overlap in these credible intervals leading to the notion that wireless clients are denied more on average than wired clients.

Now the two posterior average values will be compared. The difference between wireless posterior averages and wired posterior averages is calculated. A graph and summary statistics of the difference of the population averages will be presented.



From the density plot, it is seen that the vast majority of values are positive. This means that for the majority of the time the wireless posterior average is greater than the wired posterior average. The summary statistics depict the same relation. The mean value for the difference between the posterior averages is 4517.0 with a standard deviation of 3026.1. The probability that the difference is greater than zero (wireless posterior average greater than wired posterior average) is 97%. It is 90% certain that the true value for the difference of the posterior averages falls in the interval $(536.5, 9870.0)$. This interval does not contain 0.

Given the evidence produced by the analysis, we conclude that wireless network clients are denied more on average than wired clients, given they are both denied.

From this analysis we can answer the question of which client type is at a higher security risk. We conclude that client type wireless could be classified as being a more high risk client than wired. The implication of this conclusion might mean that the BYU security team might want to focus its efforts of buffering security for this particular client type.

One major limitation of this analysis that it does not consider the proportion of denied counts per client type. In other words, one client type might have a higher count because it usually has more clients than the other. Another limitation is that the analysis only accounts for 3 days worth of data. Our model could be improved if we continued this analysis by gathering more data over longer spans of time. Despite the limitations, this analysis is useful in that it informs security experts of higher risk client types and provides an approximated distribution for these two client type populations. Future studies might further investigate potential factors that cause clients to go to these blocked sites using a Bayesian Regression approach.

Code Appendix

```r
set.seed(11161997)
knitr::opts_knit$set(root.dir = "C:\\Users\\jon\\Documents\\School\\Bayesian\\Project")
#Load the libraries
library(dplyr)
library(tidyr)
library(kableExtra)

######################################################################################################
### Metropolis-Gibbs Sampler function and params
######################################################################################################

### Set up params

#Data
wireless <- c(32630, 10155, 8957, 7306, 4035, 4003, 3134, 3095)
wired <- c(10719, 7305, 5885, 5255, 4813, 4742, 2715, 2611, 1387, 1337)

#prior for p
alpha <- 1
beta <- 1

#prior for r
gamma <- 0.5
phi <- 0.5

### Gibbs sampler for the posterior distribution

#starting values
theta <- 0.5 #probability of being denied access
r <- 1 #just a dispersion parameter

#saving values
iters <- 10000
theta.save.wireless <- rep(0, iters)
theta.save.wired <- rep(0, iters)
r.save.wireless <- rep(0, iters)
r.save.wired <- rep(0, iters)
accept.r.wireless <- 0
accept.r.wired <- 0
```

```
metropolis_gibbs_nbinom_gamma <- function(data, alpha, beta, gamma, phi,
                                          theta, r, iters, theta.save, r.save, accept.r){

  for(i in 1:iters){

    #sample and save new value of theta
    alpha.p <- alpha + sum(data)
    beta.p <- beta  + r*length(data)
    theta <- rbeta(1, alpha.p, beta.p)
    theta.save[i] <- theta

    #sample and save new value of r (this is a metropolis random walk
    #that we will learn next Thursday)
    r.star <- rnorm(1, r, 1)
    if(r.star > 0){
        l.star <- sum(dnbinom(data, size=r.star, prob=1-theta, log=T)) +
          dgamma(r.star, gamma, phi, log=T)
        l.cur <- sum(dnbinom(data, size=r, prob=1-theta, log=T)) + dgamma(r, gamma, phi, log=T)
        if(log(runif(1)) < (l.star - l.cur)){
            r <- r.star
            accept.r <- accept.r + 1
        }
    }
    r.save[i] <- r

  }
  #Create a dictionary like list to return results
  results <- vector(mode="list", length=3)
  names(results) <- c("theta", "r", "accept")
  results[[1]] <- theta.save; results[[2]] <- r.save; results[[3]] <- accept.r
  return(results)

}

#####################################################################################################
### End Metropolis-Gibbs Sampler and params
#####################################################################################################

#####################################################################################################
### Run sampler
#####################################################################################################

### Run sampler for wireless clients
wireless.results <- metropolis_gibbs_nbinom_gamma(wireless, alpha, beta, gamma,
                                                  phi, theta, r, iters, theta.save.wireless,
                                                  r.save.wireless, accept.r.wireless)

### Run sampler for wired clients
wired.results <- metropolis_gibbs_nbinom_gamma(wired, alpha, beta, gamma, phi,
                                               theta, r, iters, theta.save.wired,
                                               r.save.wired, accept.r.wired)

#####################################################################################################
```

```
### End Run sampler
#####################################################################################################

#####################################################################################################
### Checking
#####################################################################################################

# Acceptance rate for wireless
#wireless.results$accept/iters

#want this to be around 0.42 (adjust the standard deviation in the normal distribution that samples
#the value of r.star; if this proportion is bigger than 0.6, make the standard deviation bigger,
#if the proportion is
#smaller than 0.3, make the standard deviation smaller)

# Acceptance rate for wired
#wired.results$accept/iters

### check for convergence
#Wireless
#plot(wireless.results$r, type='l')
#plot(wireless.results$theta, type='l')
#Wired
#plot(wired.results$r, type='l')
#plot(wired.results$theta, type='l')

# look at the joint distribution (NOT independent!)
#plot(wireless.results$theta, wireless.results$r)

#####################################################################################################
### End Checking
#####################################################################################################

m.wireless <- mean(wireless)
m.wired <- mean(wired)
v.wireless <- var(wireless)
v.wired <- var(wired)
i1 <- cbind("Mean Wireless" = m.wireless, "Var. Wireless" = v.wireless,
            "Mean Wired" = m.wired, "Var. Wired" = v.wired)

kable(i1, caption = "Summary of the data")

hist(wireless, freq=F, col=rgb(.1, .1, .6, .3), ylim = c(0,.0002),
     xlab="Number of times blocked", main="Counts for wireless and wired clients")
hist(wired, freq=F, col=rgb(.6, .1, .1, .3), add=T)
legend("topright", legend=c("Wireless", "Wired"), col = c("lightblue", "pink"), lwd = 2)

#prior for r
curve(dgamma(x, shape = gamma, rate = phi), xlab = "r", main = "Density of r", ylab = "Density")

#prior for theta
curve(dbeta(x, alpha, beta), xlab = expression(theta),
      main = expression(paste("Density of ", theta)), ylab = "Density")
```

```r
# Posterior Distributions for theta
plot(density(wireless.results$theta),
     expression(paste("Posterior Distribution of ", theta)),
     xlab = expression(theta), xlim = c(.9987,1), col = "red", lwd = 2)
lines(density(wired.results$theta), col = "blue", lwd = 2)
legend("topleft", legend=c("Wireless", "Wired"), col = c("red", "blue"), lwd = 2)

# Posterior Distributions - wired
plot(density(wireless.results$r), main = "Posterior Distribution of r",
     xlab = "r", xlim = c(0,6.5), col = "blue", lwd = 2, lty = 5)
lines(density(wired.results$r), col = "red", lwd = 2, lty = 5)
legend("topright", legend=c("Wireless", "Wired"), col = c("red", "blue"), lwd = 2, lty = 5)

#for wireless theta
wlt1 <- quantile(wireless.results$theta, .025)
wlt2 <- quantile(wireless.results$theta, .975)
wlt3 <- mean(wireless.results$theta)
wlt4 <- sd(wireless.results$theta)
wlt.sum <- rbind(wlt1, wlt2, wlt3, wlt4)
#for wirelss r
wlr1 <- quantile(wireless.results$r, .025)
wlr2 <- quantile(wireless.results$r, .975)
wlr3 <- mean(wireless.results$r)
wlr4 <- sd(wireless.results$r)
wlr.sum <- rbind(wlr1, wlr2, wlr3, wlr4)


#for wired theta
wt1 <- quantile(wired.results$theta, .025)
wt2 <- quantile(wired.results$theta, .975)
wt3 <- mean(wired.results$theta)
wt4 <- sd(wired.results$theta)
wt.sum <- rbind(wt1, wt2, wt3, wt4)
#for wired r
wr1 <- quantile(wired.results$r, .025)
wr2 <- quantile(wired.results$r, .975)
wr3 <- mean(wired.results$r)
wr4 <- sd(wired.results$r)
wr.sum <- rbind(wr1, wr2, wr3, wr4)

post.sum <- cbind(wlt.sum, wlr.sum, wt.sum, wr.sum)
rownames(post.sum) <- c("95% CrI low", "95% CrI Up", "Mean", "SD")
colnames(post.sum) <- c("Wireless Theta", "Wireless r", "Wired Theta", "Wired r")
#t(round(post.sum, 4))

kable(round(t(post.sum), 4), caption = "Summary Statistics for Posteriors")

#average count wireless
av.wireless <- wireless.results$r*wireless.results$theta/(1-wireless.results$theta)
#average count wired
av.wired <- wired.results$r*wired.results$theta/(1-wired.results$theta)
```

```
#plots
plot(density(av.wireless), col = "red", ylim = c(0, .00048), lwd = 2,
     main="Comparing difference in counts",
     xlab="Number of times blocked")
lines(density(av.wired), col = "blue", lwd = 2)
legend("topright", legend=c("Wireless", "Wired"), col = c("red", "blue"), lwd = 2)

#summary statistics
#wireless
avwl1 <- quantile(av.wireless, .025)
avwl2 <- quantile(av.wireless, .975)
avwl3 <- mean(av.wireless)
avwl4 <- sd(av.wireless)
avwl.sum <- rbind(avwl1, avwl2, avwl3, avwl4)
#wired
avw1 <- quantile(av.wired, .025)
avw2 <- quantile(av.wired, .975)
avw3 <- mean(av.wired)
avw4 <- sd(av.wired)
avw.sum <- rbind(avw1, avw2, avw3, avw4)

av.sum <- cbind(avwl.sum, avw.sum)
colnames(av.sum) <- c("Wireless", "Wired")
rownames(av.sum) <- c("95% CrI low", "95% CrI Up", "Mean", "SD")
#t(av.sum)

kable(t(av.sum), caption = "Summary of Results")

#diff
#plot
av.diff <- av.wireless - av.wired
plot(density(av.diff), col = "darkgreen", lwd = 2, main="Comparing difference in counts",
     xlab="Number of times blocked")

#summary statistics
prob <- mean(av.diff > 0)
low <- quantile(av.diff, .05)
up <- quantile(av.diff, .95)
m.diff <- mean(av.diff)
sd.diff <- sd(av.diff)
diff.sum <- cbind(low, up, m.diff , sd.diff,prob)
colnames(diff.sum) <- c("90% CrI Low.", "90% CrI Up.", "Mean Diff.",
                        "SD of Diff.", "Prob Wireless > Wired")
rownames(diff.sum) <- c()
#diff.sum

kable(diff.sum, caption = "Summary Statistics on the Differences")

#####################################################################################################
### Analysis and plots
#####################################################################################################

# Posterior Distributions - wireless
```

```r
plot(density(wireless.results$theta))
plot(density(wireless.results$r))
#calc average
av.wireless <- wireless.results$r*wireless.results$theta/(1-wireless.results$theta)
plot(density(av.wireless))

# Posterior Distributions - wired
plot(density(wired.results$theta))
plot(density(wired.results$r))
#calc average
av.wired <- wired.results$r*wired.results$theta/(1-wired.results$theta)
plot(density(av.wired))

#posterior difference of means
diff.av <- av.wired - av.wireless
plot(density(diff.av))
mean(diff.av <= 0)
quantile( diff.av, c(.05, .95))

### Can compare posterior predictive to the actual counts - wireless
x.dot.wireless <- rnbinom(iters, size=wireless.results$r, prob=1-wireless.results$theta)
hist(x.dot.wireless, freq=F, ylim=c(0, .0001), col=rgb(.1, .1, .6, .3))
hist(wireless, freq=F, col=rgb(.6, .1, .1, .3), add=T)

### Can compare posterior predictive to the actual counts - wired
x.dot.wired <- rnbinom(iters, size=wired.results$r, prob=1-wired.results$theta)
hist(x.dot.wired, freq=F, ylim=c(0, .0001), col=rgb(.1, .1, .6, .3))
hist(wired, freq=F, col=rgb(.6, .1, .1, .3), add=T)

### posterior predictive average of number of times denied - wireless
mean(x.dot.wireless)
#posterior predictive standard deviation of number of times an IP address is denied:
sd(x.dot.wireless)
### posterior predictive average of number of times denied - wired
mean(x.dot.wired)
#posterior predictive standard deviation of number of times an IP address is denied:
sd(x.dot.wired)

### Compare the two distributions here and other analysis

#####################################################################################################
### Analysis and plots
#####################################################################################################
#average of wireless denied, average of wired denied (both vector), subtract them, p(x>0)
```