SAND2011-1611P

A Statistical Model of Malicious Software Activity

Julie Ard & Ben Sanders

Abstract

A botnet is a collection of maliciously infected computers that are operating in cooperation with one another on behalf of the botnet owner. Common operations performed by botnets are to increase the size of a botnet (by hosting malicious webpages or attacking other computers), sending spam emails, performing DDOS attacks, or stealing important user data. To detect local computers that are participating in a botnet, we developed statistical models of infected computers in order to locate botnet participants. To do this, we performed a literature survey of the network activity of various botnets, formed an abstraction of those characteristics, and validated our model.

I. Motivation

Historically, antivirus and Intrusion Detection Systems (IDS) have always been fighting an uphill battle to generate new signatures of malicious activity. One of the biggest trends in malware is to connect back to a command and control system to perform nefarious actions on behalf of the owner of the command and control system. These pieces of malware are called botnets, and their highly connected nature means they can update themselves frequently to further avoid detection from antivirus and IDS systems. By modifying themselves frequently, antivirus systems have even greater difficulty in maintaining up to date signature databases. Antivirus companies and researchers have turned to behavioral monitoring systems to identify malicious activity as a supplement to the classic signature based systems.

Our hypothesis is that the behavioral attributes of a piece of malware differ sufficiently from the attributes of benign software such that we can derive a statistical model that will allow us to classify software with some confidence based on its behavioral activity alone.

We reviewed several model types and decided to proceed with a linear model rather than automata to define different network states primarily because the data available was more suited to the former. Additionally, we decided on a linear model based on values of variables rather than binary indication of events (e.g., age versus whether or not the subject is over 50 years of age) because it provided a finer grain of analysis.

II. Data

For our project, we needed to obtain in depth behavioral information for both a large amount of malicious and benign software. Fortunately, we have access to a Sandia National Laboratories proprietary system designed to automatically provide

malware. The Forensic Analysis Repository for Malware (FARM) is a tool that provides a web frontend to an analysis engine that will run a piece of suspected malware on a wide variety of commercial and governmental off the shelf tools and provide results to security analysts (Van Randwyk, Chiang and Lloyd). One of the features of FARM is the behavioral analysis for suspicious files. This entails monitoring a suspicious file executing in an isolated Windows environment, tracking network, filesystem, and registry behavior.

II.a. Samples

Obtaining malware was the easy part, as FARM houses a large corpus of malware from a variety of sources. We selected several thousand donated samples from Sandia's industry partners, as these samples are not considered sensitive. For benign data, we gathered Word and pdf documents, Windows system executables, and a variety of Portable Apps (PortableApps.com - Portable software for USB, portable and cloud drives).

We used 5394 total samples in our model. 4697 were known malware, and 697 were benign. One thing that was important to our analysis is to not let the large quantity of malware dominate over the smaller number of benign samples. To that end, before processing the data each time, we would take a random sampling of the malware, so that we had equal numbers of malicious and benign files to draw statistics from.

II.b. Variables

We ended up settling on 10 variables:

# of files created or modified in the C:/WINDOWS directory (excluding system32)
# of files created or modified in the C:/WINDOWS/system32 directory
# of files created or modified in the C:/Program Files directory
# of files created or modified in the C:/Documents and Settings directory
# of files created or modified in the root C:/ directory
# of registries read, created, or modified
# of DNS queries
# of tcp connections
# of http connections
# of udp connections

Although the outputs of several AntiVirus tools, a Rootkit Revealer, and Autorun detection were available, we decided to exclude these results as they would highly influence our model and diminish its ability to predict new threats.

The descriptive statistics for both malicious and benign raw data appear in the following tables.

Malicious	Max	Min	Mean	Std
WINDOWS	447	0	3.1467	26.8818
system32	48	0	0.4935	2.4674
Program	111	0	1.0089	8.0935
Docs and Set	103	0	3.3072	5.0403
C:/	613	0	8.3596	38.6070
Registries	1314	0	77.3181	186.2408
DNS	247	0	1.8473	12.6743
TCP	2	0	0.0009	0.0413
HTTP	149	0	1.5024	4.7356
UDP	17	0	0.1880	0.7854

Table 2-1: Malicious Data Descriptive Statistics by Variable

Benign	Max	Min	Mean	Std
WINDOWS	7	0	0.9971	0.5723
system32	49	0	0.1349	1.8995
Program	88	0	0.1994	3.6114
Docs and Set	23	0	3.1277	2.7122
C:/	54	0	5.0115	3.6758
Registries	109	0	0.9010	6.5985
DNS	3	0	0.0230	0.1993
TCP	4	0	0.0201	0.2265
HTTP	3	0	0.0115	0.1604
UDP	3	0	0.0100	0.1363

Table 2-2: Benign Data Descriptive Statistics by Variable

II.c. Visualization

MATLAB provides excellent tools for multivariate visualization. For visualization purposes, the variables were separated in to two basic categories, Files & Registries and Connections. The following two figures are scatterplots of these two categories for all 5394 data, both benign and malicious.

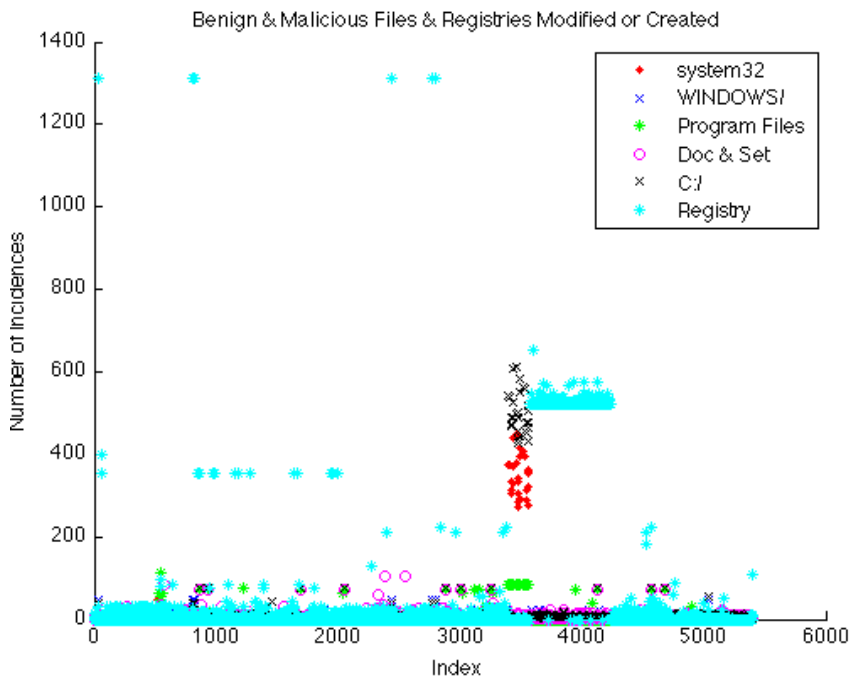

Figure 2-1: File & Registry behavior for all data

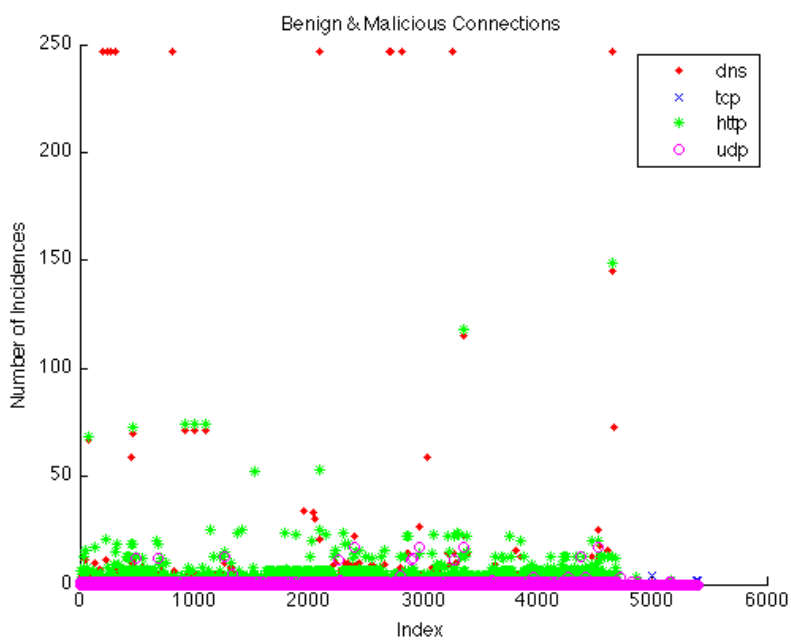

Figure 2-2: Connection behavior for all data

Now, compare the File & Registry activity between malicious and benign data. You can see, for example, that Registry creation and modification behavior is less prevalent in benign data than in malicious data.

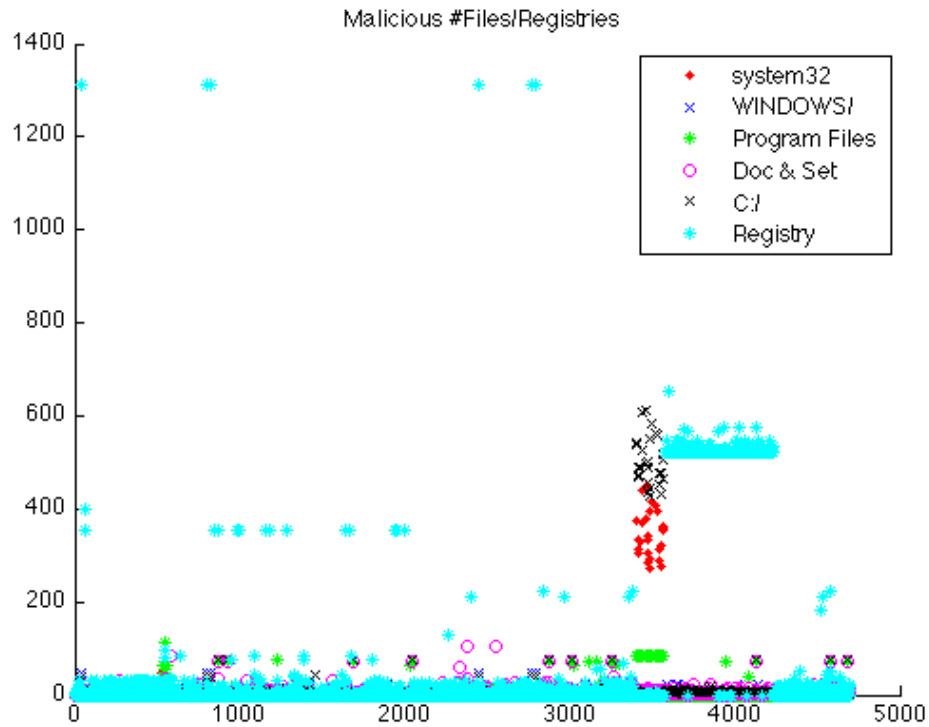

Figure 2-3: Files & Registries for malicious data

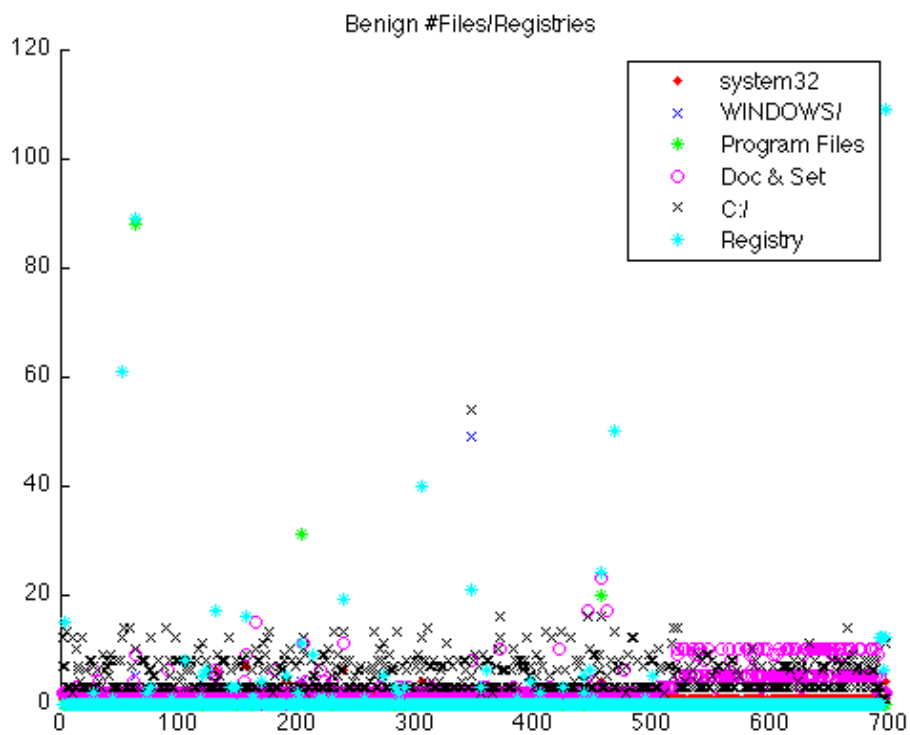

Figure 2-4: File & Registry behavior for benign data

We similarly consider benign and malicious Connection activity. DNS activity appears to be characteristic of malicious activity when all malicious data are considered. Note that a randomly selected portion of this data is used in for model training and validation.

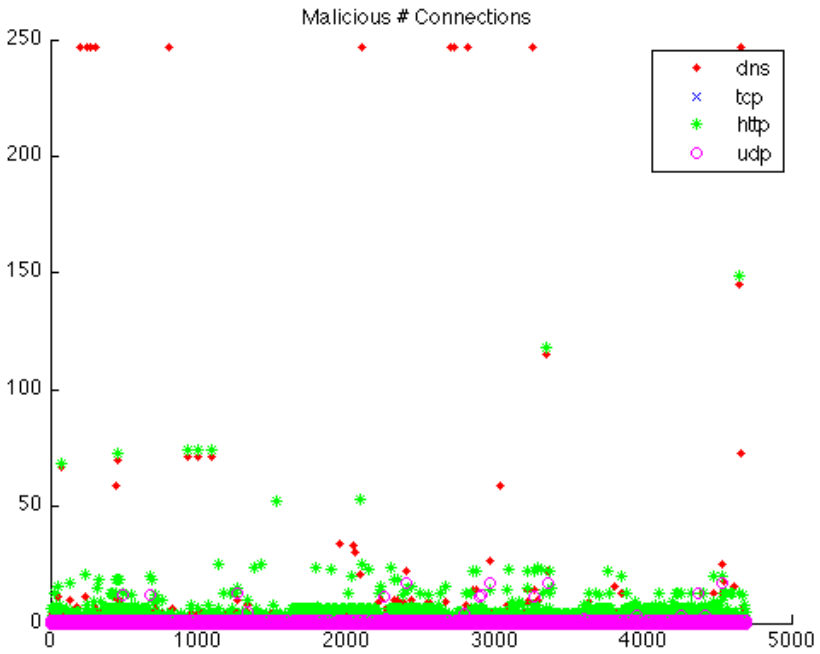

Figure 2-5: Connection behavior for malicious data

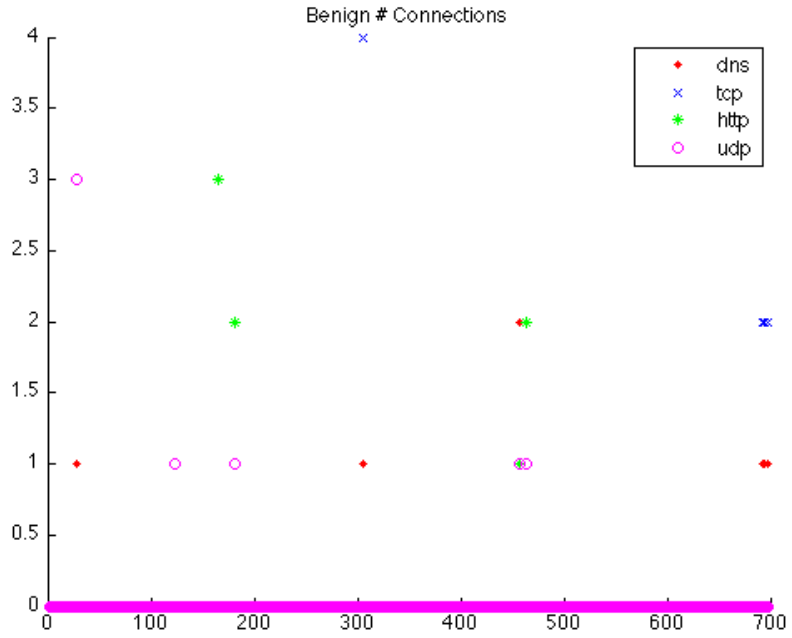

Figure 2-6: Connection behavior for benign data

III. Methodology

After data was split in to training and test data partitions, coefficients were derived using seven different models. Model validation consisted of multiplying the test data matrix with the coefficients and analyzing the test response vector.

The data is first split in to training data and test data. The extraction of test data from training data was random based on a random permutation of indices. We specified 90% of the data to be used to train the model and 10% to be used for testing. Because we have much more malicious data than benign data, early models were biased towards malicious scoring, so we constrained the script to use the same number of records for both malware and benign data. Accordingly, all benign data was used with the same number of further randomly selected malicious data. For ease of analysis, the test data consists of two partitions of random sets of known malicious and benign files. This does not affect the validation results since "testing" consists simply of matrix multiplication between each test data row and the coefficient column.

Model Discussion

We built seven behavior models. The first uses multilinear regression, and the others are variations on a generalized linear model using the Normal, Binomial, and Poisson distributions, both with and without constant coefficients. Gamma and Inverse Gaussian distributions were also available but would not run because the response vector may include negative responses. For all models, the vector of predicted responses consisted of 1's for known malware and 0's for benign files.

The multilinear regression model outputs coefficients for each variable using multilinear regression. If $X_1,\ldots,X_{10}$ are the data vectors for each variable and Y is the predicted response vector, the linear regression model will output coefficients $b_1,\ldots,b_{10}$ such that

$$b_1*X_1 + b_2*X_2 + \ldots + b_{10}*X_{10} = Y$$

The generalized linear models output coefficients for each variable (plus a constant term, if used) for a generalized linear regression using the specified probability distribution. We used the normal, binomial, and Poisson distributions, both with and without constant coefficients. In the case that constant coefficients were not used, the above relation is accurate but with the derivation of the coefficients differing according to the distribution. In the cases that constant coefficients were used, if $X_1,\ldots,X_{11}$ are the data vectors for each variable and Y is the predicted response vector, the linear regression model will output coefficients $b_0,b_1,\ldots,b_{11}$ such that

$$b_0 + b_1*X_1 + b_2*X_2 + \ldots + b_{11}*X_{11} = Y$$

In summary:

Model ID	Description	Distribution	Constant Coeff
MLR	Multilinear Regression	n/a	No
GLMN	Generalized Linear Model	Normal	No
GLMB	Generalized Linear Model	Binomial	No
GLMP	Generalized Linear Model	Poisson	No
GLMNC	Generalized Linear Model	Normal	Yes
GLMBC	Generalized Linear Model	Binomial	Yes
GLMPC	Generalized Linear Model	Poisson	Yes

Table 2-1: Model Summary

III.a. Multilinear Regression Model

As mentioned above, this model produces one coefficient per variable. Using 1256 total randomly selected data records in the training set (with 628 malicious and 628 benign), the coefficients were:

Variable	Coefficients (MLR)
C:/WINDOWS	-0.0620
C:/WINDOWS/system32	-0.0445
C:/Program	-0.0112
C:/Documents and Settings	0.0069
C:/WINDOWS	0.0484
Registries	0.0012
DNS	0.0039
TCP	-0.0721
HTTP	0.0733
UDP	0.1093

Table 3-1: MLR Coefficients

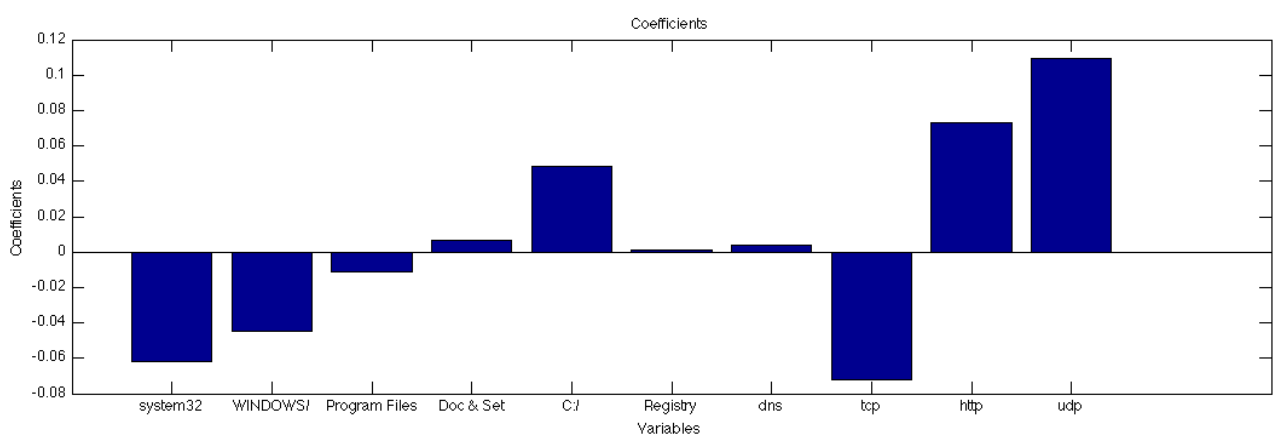

Figure 3-1: MLR Coefficients

As you can see, the TCP and UDP variables are most influential while the Registry and DNS variables influence this model the least. The negative coefficients, System32, WINDOWS, Program Files, and TCP suggest that they indicate benign behavior.

The coefficients were then tested by multiplying them with the test data, which consisted of 70 malicious records and 70 benign records.

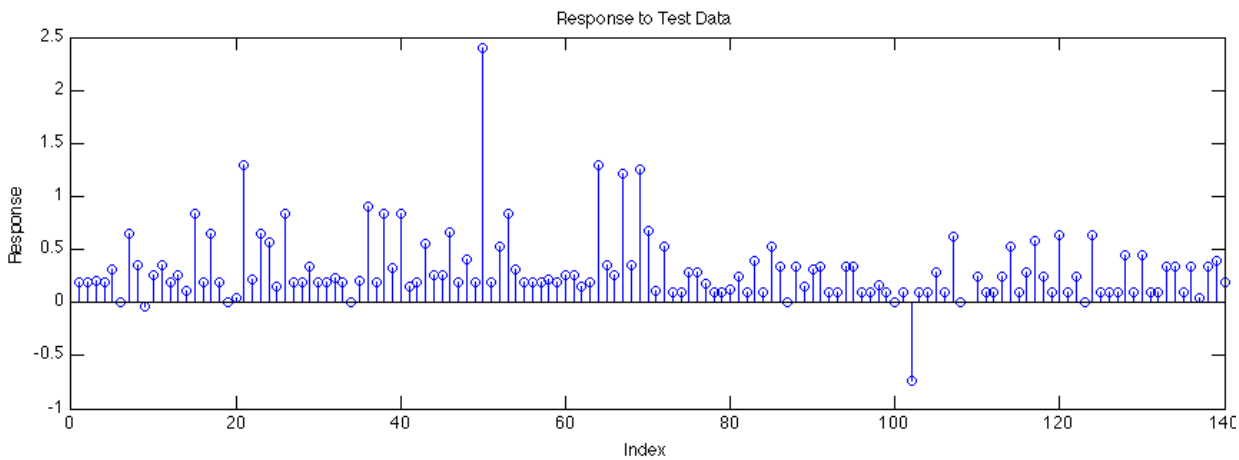

Figure 3-2: MLR Response

In general, the response validates the malicious and benign data. The first half of the test data (1-70) was malicious, while the second half (71-140) was benign. The basic descriptive statistics confirm this.

MLR	Malicious Responses	Benign Responses
Max	2.3971	0.6295
Min	-0.040	-0.7386
Mean	0.4014	0.20715
Std	0.3981	0.20308

Table 3-2: MLR Response Descriptive Statistics

III.b. Generalized Linear Model with Normal distribution

This model also generates one coefficient per variable. Using the same 1256 total data records as above in the training set (with 628 malicious and 628 benign), the coefficients were:

Variable	Coefficients (GLMN)
C:/WINDOWS	-0.0620
C:/WINDOWS/system32	-0.0445
C:/Program	-0.0112
C:/Documents and Settings	0.0069
C:/WINDOWS	0.0484
Registries	0.0012
DNS	0.0039
TCP	-0.0721
HTTP	0.0733
UDP	0.1093

Table 3-3: GLMN Coefficients

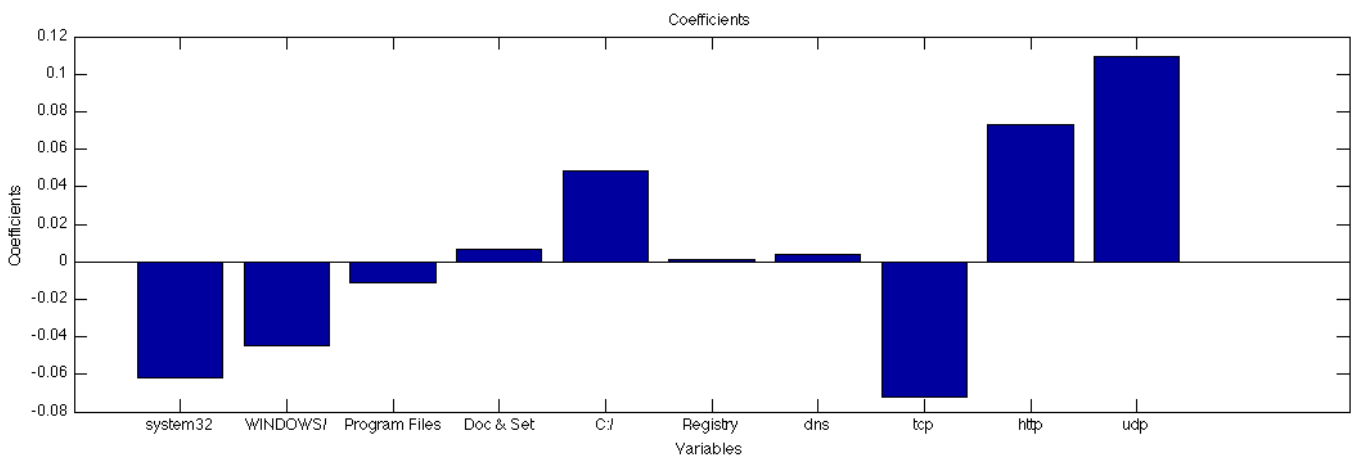

Figure 3-4: GLMN Coefficients

These coefficients are very similar to those for Multilinear Regression, and therefore, TCP and UDP variables are similarly most influential while the Registry and DNS variables similarly influence this model the least. The same coefficients, System32, WINDOWS, Program Files, and TCP are negative which suggests that they indicate benign behavior.

These coefficients are so similar that it leads us to believe that computational methods between the Generalized Linear Regression – Normal and Multilinear Regression functions are very similar. MATLAB documentation does not contain detailed information on this, but does state that residuals of the observed responses

have a normal distribution. The default distribution for the Generalized Linear Regression function is the Normal distribution.

The coefficients were then tested by multiplying them with the test data, which consisted of 70 malicious records and 70 benign records.

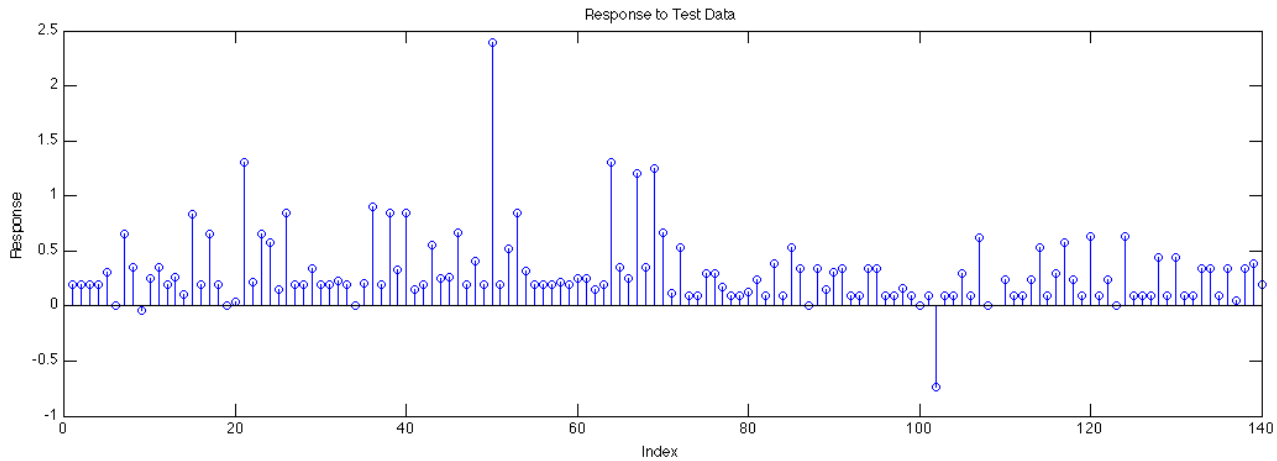

Figure 3-5: GLMN Response

In general, the response validates the malicious and benign data. The first half of the test data (1-70) was malicious, while the second half (71-140) was benign. The basic descriptive statistics confirm this.

GLMN	Malicious Responses	Benign Responses
Max	2.39715	0.6294
Min	-0.0401	-0.7386
Mean	0.4014	0.2071
Std	0.3981	0.2030

Table 3-4: GLMN Response Descriptive Statistics

We notice that all of these statistics are the same as those generated by Multilinear Regression within about 10 significant digits.

III.c. Generalized Linear Model with Binomial distribution

This model also produces one coefficient per variable. Using the same 1256 total data records in the training set (with 628 malicious and 628 benign), the coefficients were:

Variable	Coefficients (GLMB)
C:/WINDOWS	-0.0234
C:/WINDOWS/system32	-0.0851
C:/Program	0.0725
C:/Documents and Settings	-0.2032
C:/WINDOWS	0.0034
Registries	0.0226
DNS	2.2020
TCP	-45.0792
HTTP	0.4093
UDP	0.0522

Table 3-5: GLMB Coefficients

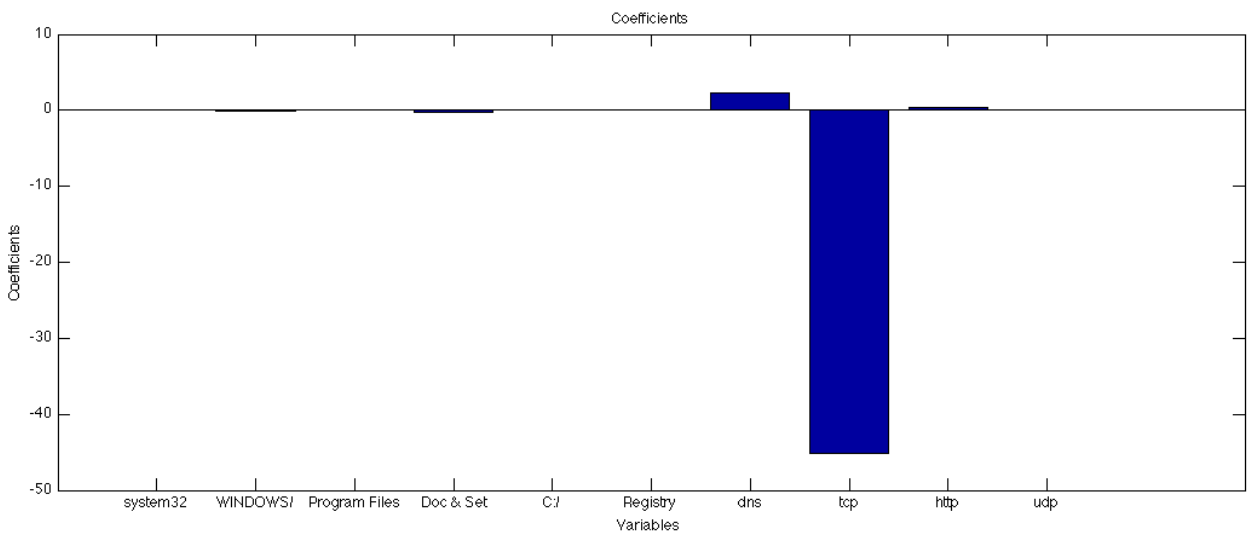

Figure 3-7: GLMB Coefficients

Here we see some very different results from the previous two models. We note that TCP and DNS are the major contributors to the response. The smallest contributor is WINDOWS. WINDOWS, System32, Docs&Sets, and TCP are negative, associating them with benign behavior.

The coefficients were then tested by multiplying them with the test data, which consisted of 70 malicious records and 70 benign records.

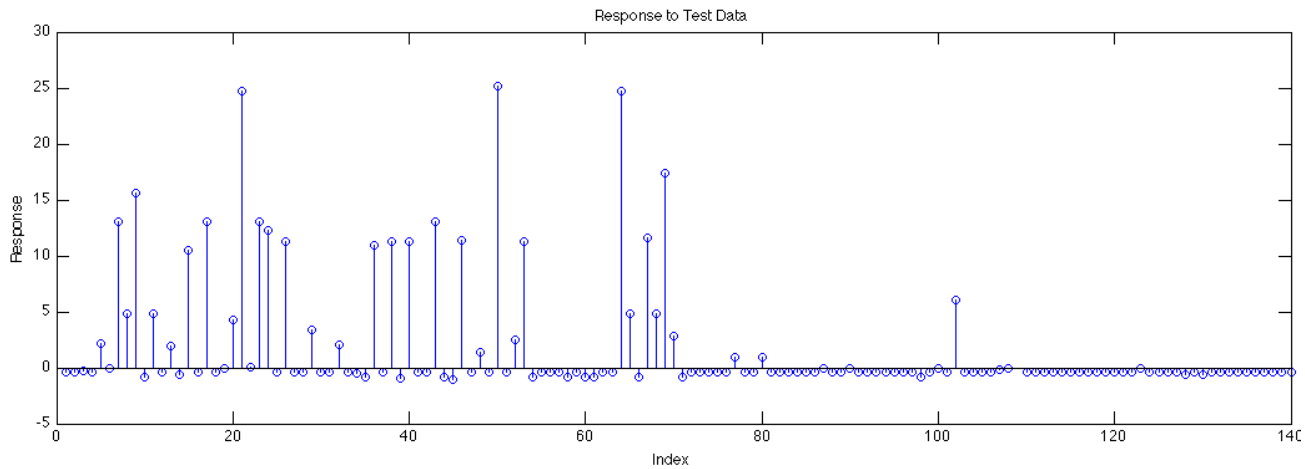

Figure 3-8: GLMB Response

Here we see a more obvious validation of the malicious and benign data. The first half of the test data (1-70) was malicious, while the second half (71-140) was benign. The basic descriptive statistics confirm this.

GLMB	Malicious Responses	Benign Responses
Max	25.1585	6.0974
Min	-1.0189	-0.8259
Mean	4.0253	-0.2612
Std	6.8844	0.8209

Table 3-6: GLMB Response Descriptive Statistics

These results are radically different from the previous two models. We note that the standard deviation for malicious responses is very high.

III.d. Generalized Linear Model with Poisson distribution

This model is the last analyzed which produces one coefficient per variable. Using the same 1256 total data records in the training set (with 628 malicious and 628 benign), the coefficients were:

Variable	Coefficients (GLMP)
C:/WINDOWS	0.1036
C:/WINDOWS/system32	0.0336
C:/Program	0.0176
C:/Documents and Settings	-0.1078
C:/WINDOWS	-0.0797
Registries	0.0010
DNS	0.0022
TCP	-44.5639
HTTP	0.0704
UDP	0.1354

Table 3-7: GLMP Coefficients

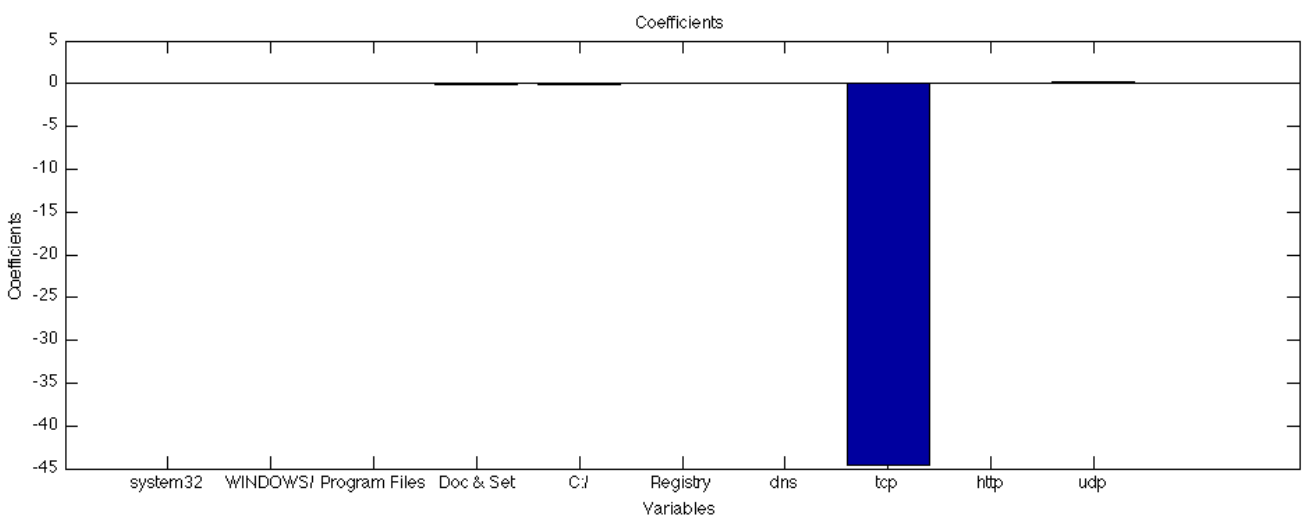

Figure 3-10: GLMP Coefficients

The coefficients are very similar to those generated by the GLM-Binomial model, with TCP activity dominating the response. This time, UDP is the second greatest contributor while Registries impacts the response the least. WINDOWS, Docs&Sets, and of course TCP are negative, associating them with benign behavior.

The coefficients were then tested by multiplying them with the test data, which consisted of 70 malicious records and 70 benign records.

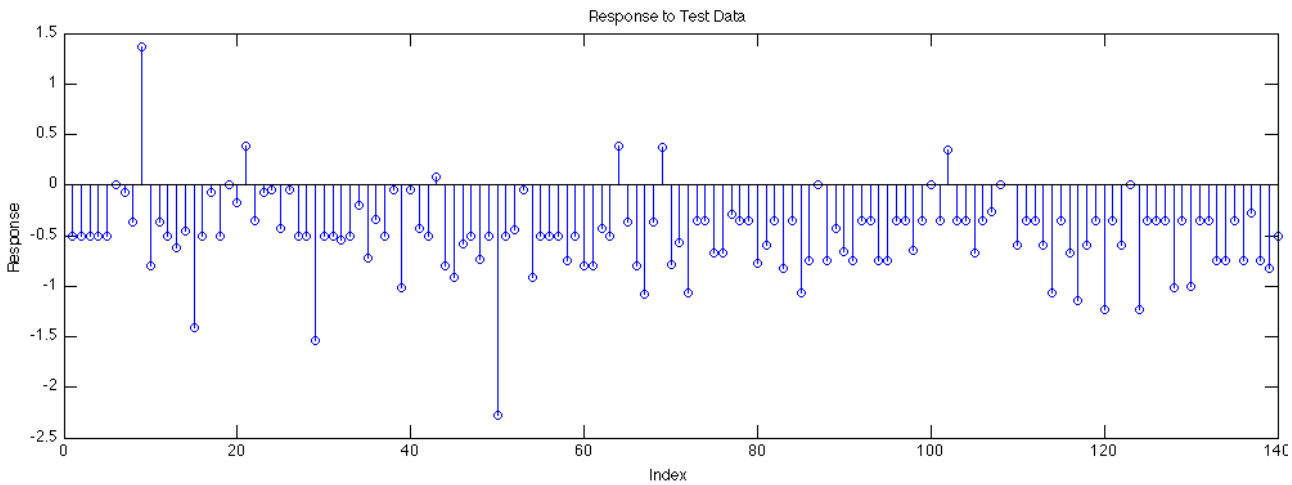

Figure 3-11: GLMP Response

Here we see a more obvious validation of the malicious and benign data. The first half of the test data (1-70) was malicious, while the second half (71-140) was benign. The basic descriptive statistics confirm this.

GLMP	Malicious Responses	Benign Responses
Max	1.36905	0.3508
Min	-2.2754	-1.2274
Mean	-0.4612	-0.5274
Std	0.4687	0.3080

Table 3-8: GLMP Response Descriptive Statistics

These results less extreme behavior than the GLM-Binomial model with the same emphasis on TCP behavior.

III.e. Generalized Linear Model with Normal distribution and a Constant Coefficient

This model generates eleven total coefficients, one for each variable plus a constant coefficient. Using the same 1256 total data records as above in the training set (with 628 malicious and 628 benign), the coefficients were:

Variable	Coefficients (GLMNC)
Constant	0.4369
C:/WINDOWS	-0.0022
C:/WINDOWS/system32	-0.0108
C:/Program	0.0054
C:/Documents and Settings	-0.0118
C:/WINDOWS	0.0016
Registries	0.0009
DNS	0.0027
TCP	-0.1748
HTTP	0.0615
UDP	0.1007

Table 3-9: GLMNC Coefficients

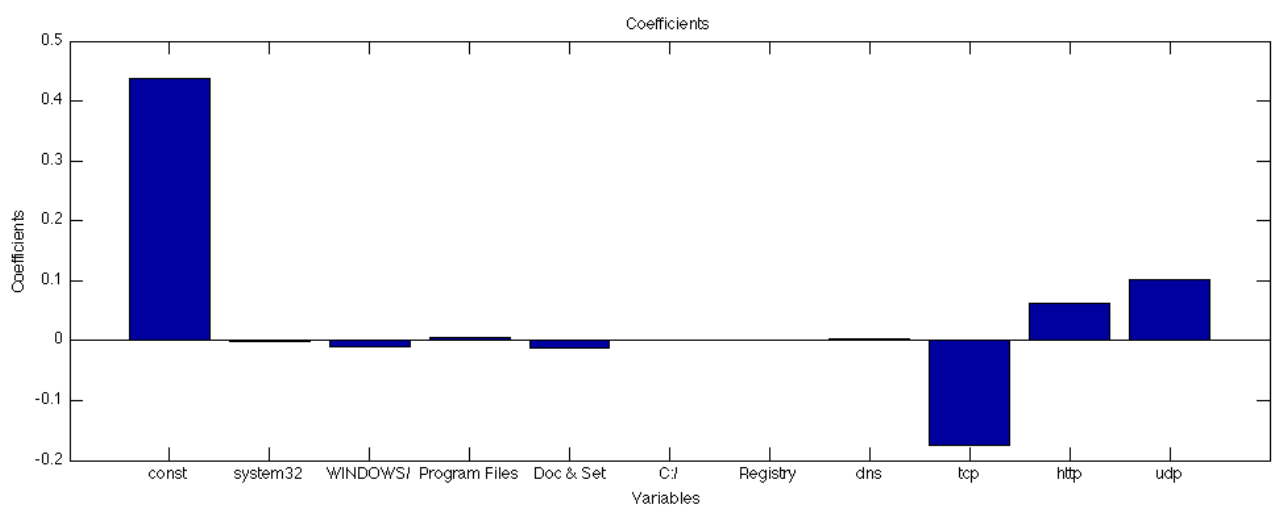

Figure 3-13: GLMNC Coefficients

These coefficients reflect the GLM-Normal coefficients above with the emphasis on TCP and UDP variables. However, we can see that the constant term is the most in magnitude.

The coefficients were then tested by multiplying them with the test data, which consisted of 70 malicious records and 70 benign records.

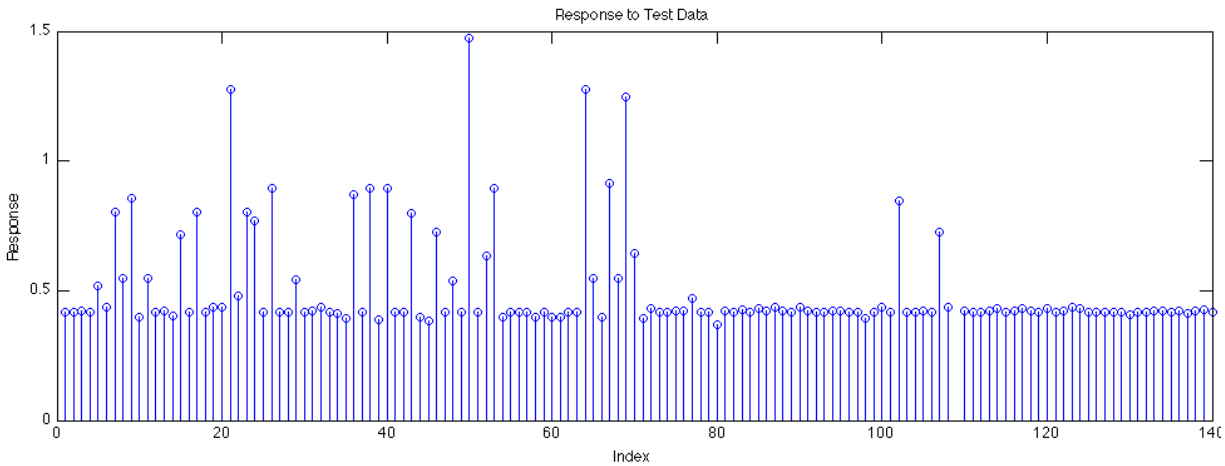

Figure 3-14: GLMNC Response

In general, the response validates the malicious and benign data. The first half of the test data (1-70) was malicious, while the second half (71-140) was benign.

GLMNC	Malicious Responses	Benign Responses
Max	1.4751	0.8454
Min	0.3851	0.3712
Mean	0.5694	0.4308
Std	0.2504	0.0637

Table 3-10: GLMNC Response Descriptive Statistics

We notice that all of the responses are biased due to the constant term (0.43) such that the minimum score for both malicious and benign data are almost the same. The benign and malicious means are also very similar.

III.f. Generalized Linear Model with Binomial distribution with a Constant Coefficient

This model generates eleven total coefficients, one for each variable plus a constant coefficient. Using the same 1256 total data records as above in the training set (with 628 malicious and 628 benign), the coefficients were:

Variable	Coefficients (GLMBC)
Constant	-0.1727
C:/WINDOWS	-0.0325
C:/WINDOWS/system32	-0.0851
C:/Program	0.0494
C:/Documents and Settings	-0.1843
C:/WINDOWS	0.0216
Registries	0.0263
DNS	2.1409
TCP	-45.1588
HTTP	0.3771
UDP	0.0675

Table 3-11: GLMBC Coefficients

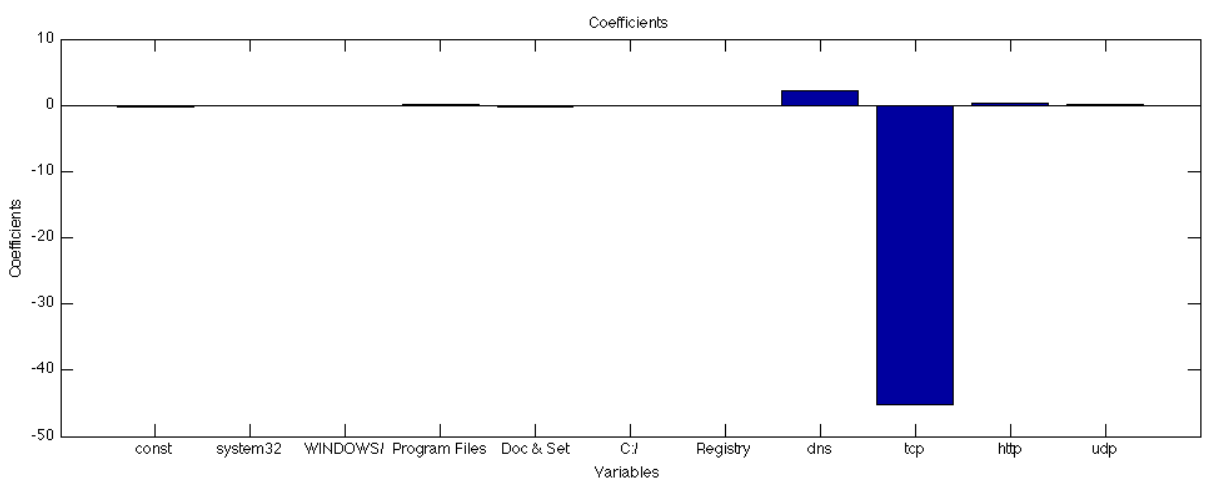

Figure 3-16: GLMBC Coefficients

The constant coefficient in the GLM-Binomial model is much smaller but recalls the high dependence on the TCP variable and second-highest dependence on DNS.

The coefficients were then tested by multiplying them with the test data, which consisted of 70 malicious records and 70 benign records.

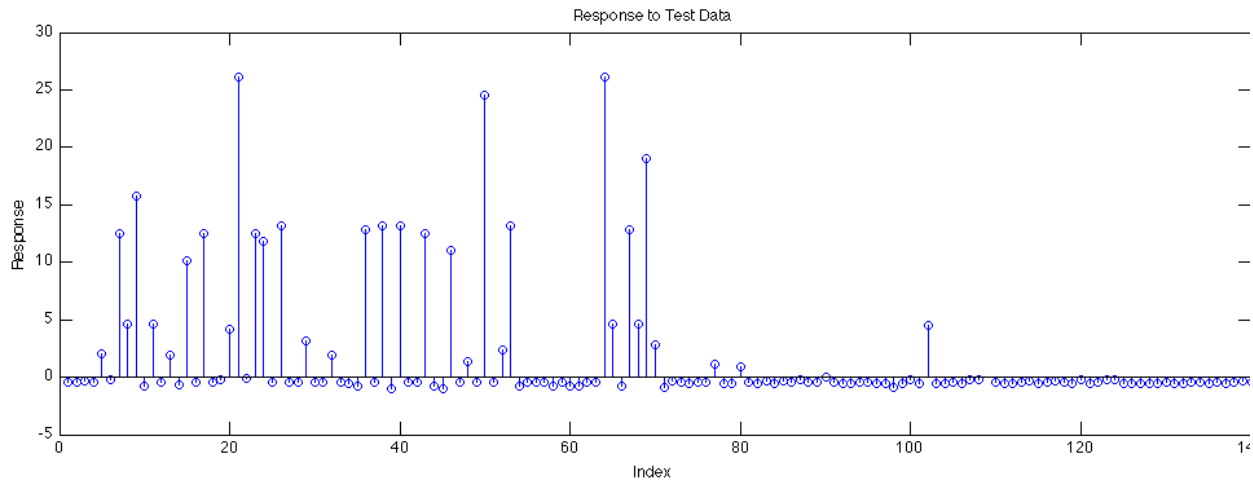

Figure 3-17: GLMBC Response

In general, the response validates the malicious and benign data. The first half of the test data (1-70) was malicious, while the second half (71-140) was benign.

GLMBC	Malicious Responses	Benign Responses
Max	26.1577	4.5505
Min	-0.9970	-0.8775
Mean	4.1350	-0.3240
Std	7.1758	0.6564

Table 3-12: GLMBC Response Descriptive Statistics

The minimum score for both malicious and benign data are again very close. The standard deviation for malicious responses is very high.

III.g. Generalized Linear Model with Poisson distribution and a Constant Coefficient

This model also generates eleven total coefficients, one for each variable plus a constant coefficient. Using the same 1256 total data records as above in the training set (with 628 malicious and 628 benign), the coefficients were:

Variable	Coefficients (GLMPC)
Constant	-0.8071
C:/WINDOWS	-0.0013
C:/WINDOWS/system32	-0.0180
C:/Program	0.0077
C:/Documents and Settings	-0.0220
C:/WINDOWS	0.0013
Registries	0.0013
DNS	0.0039
TCP	-44.3407
HTTP	0.0775
UDP	0.1269

Table 3-13: GLMPC Coefficients

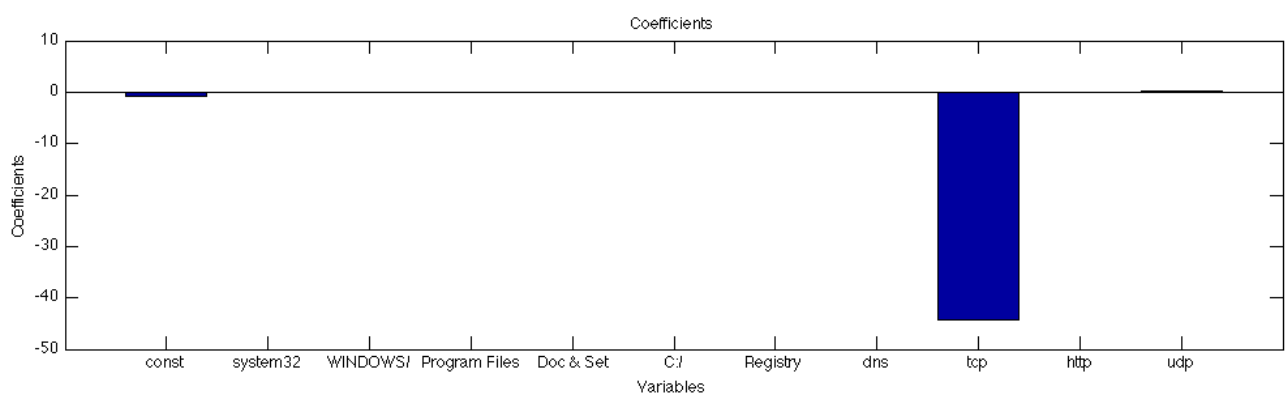

Figure 3-19: GLMPC Coefficients

Unlike the GLM-Poisson model without the constant term, this model reflects the extreme dependence on TCP that both Binomial models have, with a smaller constant coefficient.

The coefficients were then tested by multiplying them with the test data, which consisted of 70 malicious records and 70 benign records.

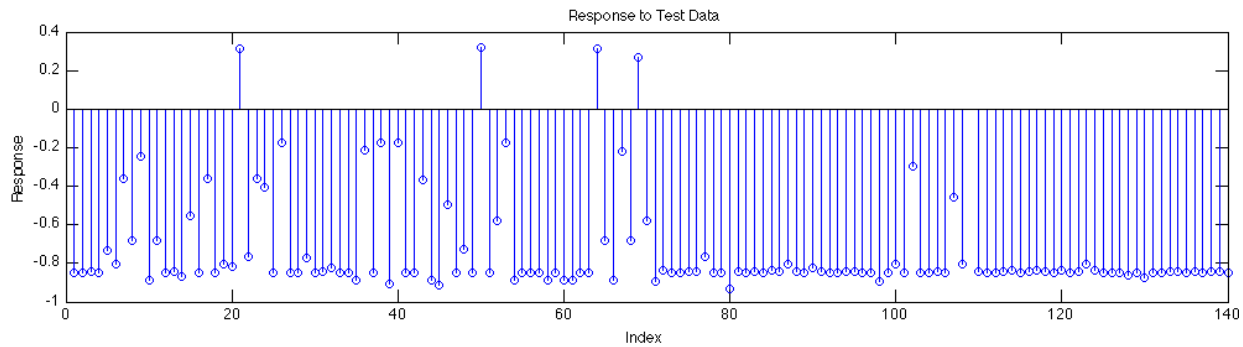

Figure 3-20: GLMPC Response

In general, the response validates the malicious and benign data. The first half of the test data (1-70) was malicious, while the second half (71-140) was benign.

GLMPC	Malicious Responses	Benign Responses
Max	0.3188	-0.2941
Min	-0.9109	-0.9328
Mean	-0.6551	-0.8314
Std	0.3262	0.0826

Table 3-14: GLMPC Response Descriptive Statistics

Here most of the responses are negative. The minimum score for both malicious and benign data is again almost the same, with the means very similar.

IV. Conclusions

Though our methodology did show promise in being able to positively identify malicious software, there are obviously a few areas in which improvements could be made.

One of the issues we noticed with the benign software is that it is fairly inactive without user intervention. Though many commonplace pieces of software may be capable of making filesystem changes and network activity comparable to any average piece of malware, the reality is that most benign software only performs those actions after being instructed to do so by the user in some way. For example, the portable apps required installation and extraction prior to doing anything else. On the other hand, malware is known for not requiring any user intervention, and will quickly root itself in the filesystem and registry, and possibly start contacting other bots. What this means for our model is that it is good at detecting samples that are performing a variety of forensically significant actions autonomously (such as when run inside of FARM), but it will not correctly classify software on a live network that has user intervention to direct it.

Both Binomial variations produced the most favorable results by recognizing the difference between malicious and benign data, although many malicious results were also scored very low (as benign). The Binomial models are also the only

models to assign DNS a relatively high coefficient, recognizing our early observation that DNS queries were higher among malicious data. However, the extreme behavior of these models are of concern because they may not perform well outside of our test environment. The TCP coefficient is so high that dependence on other variables is very low by comparison. The high standard deviations for malicious data is of note but may also be a result of the high magnitude of the coefficients. Both variations of the Binomial model also produced the most extreme Max response values for both malicious and benign data.

The high negative value of 4 of 7 models' TCP coefficients can be attributed to the fact that the maximum malware TCP value was 2 and the maximum benign TCP value was 4. Since these values are quite low as compared to other variables, and few variables are associated with benign software, this coefficient was made to be quite influential in some models in order to meet the benign predicted response of 0.

Models using constant coefficients produced more ambiguous descriptive statistics than models which did not have a constant coefficient. The use of constant coefficients may not be the best idea in cases where our observed responses are artificial. For example, a response of 1 was generated for malicious training data and a response of 0 was generated for benign training data. In early model development when the malicious data far outnumbered the benign data, this caused a significant bias in favor of malicious data because the constant coefficient was higher. When the expected output mostly consisted of ones, the constant coefficient adjusted
More realistic scores for training data, or perhaps variable scores based on the maliciousness of the malicious file, is an area of future consideration. One possible near-term solution to this problem is randomly generated scores between, for example, 0 and 0.5 for benign data and 0.5 and 1 for malicious data. This issue may also be solved with data centering and scaling in future work.

V. Future Work

Future work will include more variables and string analysis so that non-numeric data can be analyzed. The ability to accommodate variable-length datasets would be desirable. For example, the connection data includes port numbers for each connection. Currently this and other data is not in our model because one record might contain four tcp connections with four sets of source and destination port numbers, while another record may contain one tcp connection with one set of port numbers.

More models are available to explore and, as we have seen in this project, an option such as the probability distribution significantly affects results. As mentioned above, future work would examine assignment of scores used in the malicious and benign training data.

Works Cited

Arbor Networks, <http://www.arbornetworks.com/>.
Offensive Computing, <http://www.offensivecomputing.net/>.
PortableApps.com - Portable software for USB, portable and cloud drives. Rare Ideas, LLC. 6 3 2011 <http://portableapps.com>.
Van Randwyk, Jamie, et al. "Farm: An automated malware analysis environment." 42nd Annual IEEE International Carnahan Conference on Security Technology. IEEE ICCST, 2008. 321-325.