An Empirical Study of Reflection Attacks Using NetFlow Data

Edward Chuah[1*] and Neeraj Suri[2]

[1*]The University of Aberdeen, Aberdeen, AB24 3FX, UK.
[2]Lancaster University, Lancaster, LA1 4YW, UK.

*Corresponding author(s). E-mail(s): thuan.chuah@abdn.ac.uk;
Contributing authors: neeraj.suri@lancaster.ac.uk;

Abstract

Reflection attacks are one of the most intimidating threats organizations face. A reflection attack is a special type of distributed denial-of-service attack that amplifies the amount of malicious traffic by using reflectors and hides the identity of the attacker. Reflection attacks are known to be one of the most common causes of service disruption in large networks. Large networks perform extensive logging of NetFlow data, and parsing this data is an advocated basis for identifying network attacks. We conduct a comprehensive analysis of NetFlow data containing 1.7 billion NetFlow records and identified reflection attacks on the Network Time Protocol (NTP) and NetBIOS servers. We set up three regression models including the Ridge, Elastic Net and LASSO. To the best of our knowledge, there is no work that studied different regression models to understand patterns of reflection attacks in a large network. In this paper, we (a) propose an approach for identifying correlations of reflection attacks, and (b) evaluate the three regression models on real NetFlow data. Our results show that (a) reflection attacks on the NTP servers are not correlated, (b) reflection attacks on the NetBIOS servers are not correlated, (c) the traffic generated by those reflection attacks did not overwhelm the NTP and NetBIOS servers, and (d) the dwell times of reflection attacks on the NTP and NetBIOS servers are too small for predicting reflection attacks on these servers. Our work on reflection attacks identification highlights recommendations that could facilitate better handling of reflection attacks in large networks.

Keywords: Large networks, NetFlow data, Reflection attacks, Regression analysis

1 Introduction

The capacity of large networks has grown significantly in order to sustain the level of performance required by machine learning, scientific and engineering applications. In this context, the security of the network has become critical to meet the expectations of its users. Analyzing network attacks requires awareness of the sequence of events encountered by the network component. While recent works have focused on analyzing attacks on specific network components [1, 2], answering how an attack on a network occurs requires an integrated approach towards correlation-based log mining [3, 4]. Correlation analysis has been widely used to detect intrusions in large networks [5–9], with its strengths in aggregating several alerts with low false positives and low false negatives, resulting in tremendous improvements in detection accuracy. A recent study empirically evaluated the Pearson and Spearman-Rank correlation algorithms using NetFlow data obtained from an enterprise network [10]. From their study, they observed that reflection attacks on the Secure Shell (SSH) and Domain Name Service (DNS) servers exist in the NetFlow data and those attacks are not correlated.

Several recent large-scale Distributed Denial-of-Service (DDoS) attacks studies have provided valuable insights into DDoS attacks [11–14]. These studies have shown that DDoS attacks are regularly executed on many network protocols. In a DDoS attack, a large volume of network packets is generated to flood a target host without using an intermediary. In contrast to DDoS attacks, which do not mask the sender's source IP address, a reflection attack is a special type of DDoS attack that uses any TCP or UDP-based service as a reflector and masks the sender's source IP address [15]. A spoofed network packet, where the source IP address is replaced by the IP address of another device, is typically used to send the response to the victim. Thus, an attacker can magnify the amount of malicious traffic and obscure the sources of the attack traffic to cause significant disruption to the operation of a large network. As such, it is important to identify correlations of reflection attacks, as it is to identify the dwell time between these attacks. We define the dwell time as the time elapsed between the start time of one reflection attack and the start time of the next reflection attack. When network attack prediction schemes are supported by knowledge of the dwell times of an attack, it can help the network administrators in using network attack mitigation schemes to respond to an impending attack [16]. When the dwell times of an attack are small, a network attack mitigation scheme which scatters the attack traffic can be used to absorb the attack.

Several recent works have developed Pearson correlation-based methods that identified DDoS attacks [17], identified reflection attacks [10], detected activities of groups of bots [18], and detected network intrusions [19]. S. Chawla *et al.* [17] proposed a framework that used Pearson correlation to identify DDoS attacks and flash events. D.P. Hostiadi and T. Ahmad [18] proposed a new model that detected correlations of activities of group of bots. Their model consists of four phases: (a) data preprocessing, (b) data segmentation, (c) feature

extraction and (d) bot group detection. They implemented (a) the Mean Absolute Error metric that measures the similarity of activities between two groups of bots, and (b) the Pearson correlation algorithm that finds relationships between the activities of two bot groups. A. Heryanto *et al.* [19] implemented a feature selection workflow that uses Pearson correlation to identify important network metrics for detecting intrusions. E. Chuah *et al.* [10] applied Pearson correlation and Spearman-Rank correlation to identify the dates of reflection attacks. Although these works showed that the Pearson correlation algorithm can identify relationships between malicious activities, it has some limitations that we address in this paper. First, Pearson correlation only identifies relationships between two samples. Second, several correlated samples can be produced and all the correlated samples must be manually analyzed before an attack can be identified, which is not desirable because it is a time-consuming process that incurs a significant delay in identifying correlations of a network attack. Therefore, we use the power of regression models, which belong to a type of supervised learning that learns a relationship between a dependent variable and multiple independent variables. We train the Ridge, Elastic Net and LASSO regression models on NetFlow data to obtain the regression coefficients for all independent variables, and determine the applicability of these regression models in identifying correlations of reflection attacks.

In this paper, we conduct an empirical analysis of reflection attacks in a large enterprise network, carefully compare the Ridge, LASSO and Elastic Net regression models and present several new findings. The correlation of reflection attacks on the NTP server and correlation of reflection attacks on the NetBIOS server are new ones and have not been reported in an earlier paper [10]. We validate our approach on 1.7 billion NetFlow records obtained from a large enterprise network operated by Los Alamos National Laboratories, and apply statistical validation methods to ensure that the results are accurate. The main contributions of this paper are given as follows:

- We identify reflection attacks in a large enterprise network and provide estimates of NetFlow records which are not correlated with the reflection attack.
- We analyze the NetFlow records which are associated with a reflection attack to drill down into their specific activity. Based upon the insights gained from our correlation analysis, we discuss how these findings can be used to improve the network's security against reflection attacks.
- We extract the NetFlow records associated with the reflection attack and obtain their dwell times.

Our initial assumption is that reflection attacks are correlated in the NetFlow data. We compared the Ridge, Elastic Net and LASSO regression models and are surprised to learn that the regression coefficients learned by all three regression models are close to 0 or equal to 0. Furthermore, the dwell times of reflection attacks ranged from 0 to 198 seconds, multiple source and destination devices were associated with reflection attacks on the NTP and

NetBIOS servers, and a small percentage of network traffic was generated by the reflection attack.

The remainder of this paper is organized as follows: First, we review the related works in Section 2. Then, we describe the network model and NetFlow data in Section 3. We present the motivation and describe the details of our approach in Section 4. Our evaluation on the NetFlow data obtained from an enterprise network is presented in Section 5. We discuss the results and limitations of our approach in Sections 6 and 7 respectively, and we conclude with a summary and future work in Section 8.

2 Related work

We divide the related works into two categories: (a) machine learning-based intrusion detection systems that detected DDoS attacks, and (b) correlation analysis-based intrusion detection systems that detected DDoS attacks.

2.1 Machine Learning-based Intrusion Detection Systems

We focused on very recent works that developed Intrusion Detection Systems (IDS) which integrated machine learning techniques to detect DDoS attacks. In [20], the authors proposed a natural language processing (NLP)-based approach called *DDoS2Vec* that learns the characteristics of DDoS attacks. They evaluated their approach on one year's worth of flow samples obtained from an Internet Exchange Provider and compared the performance of DDoS2Vec with Word2Vec, Dos2Vec and Latent Semantic Analysis. In [21], the authors proposed a novel approach that stacks multiple deep neural networks (DNN) to detect DDoS attacks. They evaluated their approach on a benchmark Cybersecurity dataset and compared the performance of their method with existing machine learning models. In [22], the authors implemented an approach that compared multiple Support Vector Machine (SVM) kernels that are trained with uncorrelated features to detect reflection amplification DDoS attacks on the Simple Network Management Protocol (SNMP) and DNS servers. In [23], the authors proposed a feature reduction method that integrated Information Gain (IG) and correlation-based feature selection techniques to detect reflection and standard DDoS attacks. They evaluated their method on two public Cybersecurity datasets and compared the performance of their approach with state-of-the-art feature selection methods. In [24], the authors developed a decision tree-based IDS that uses the J48 classifier to detect reflection amplification DDoS attacks, and evaluated their method on the CICDDoS2019 Cybersecurity dataset. In [25], the authors proposed a novel technique that combined clustering and classification machine learning algorithms. Their technique consists of three phases. In the first phase, the DBSCAN clustering algorithm is used to separate DDoS traffic from normal traffic. In the second phase, the Euclidean distance metric is used to calculate the features in each cluster. In the third phase, a classification model is built and a label that indicates whether a cluster contains DDoS traffic or normal

traffic is assigned to each cluster. They evaluated their method on two public Cybersecurity datasets and compared the performance of their classifier with the Decision Tree, Random Forest, Naive Bayes and Support Vector Machine classifiers. In [26], the authors compared the Random Forest (RF), Naive Bayes (NB), Logistics Regression (LR), K-Nearest Neighbour (KNN) and Multilayer Perceptron (MLP) algorithms to filter normal traffic from DDoS traffic, and evaluated these algorithms on two public DDoS datasets. In [27], the authors proposed a deep learning model to detect DDoS attacks. They evaluated their model on the CICDDoS2019 dataset that includes traces of DDoS attacks on several network protocols, and showed that a three layer deep neural network model achieved the highest detection accuracy. Further, in [28], the authors proposed an approach called *MOP-IDS*. It is based on a Multi-Objective Optimization (MOO) process composed of: (a) clustering alerts generated by multiple IDS to decrease the set of alerts, (b) filtering alerts to create a set of potential false alarms, (c) grouping similar alerts produced by the different IDS and (d) classifying an alert as a false positive or false negative. The performance of MOP-IDS was evaluated using accuracy, true positive rate and true negative rate metrics on three public Cybersecurity datasets that contain denial-of-service traces . In [29], the authors proposed a novel approach called *CANN*. It is based on K-Means clustering that sums two distances: (a) the distance between a data sample and its cluster center and (b) the distance between a data sample and its nearest neighbour. A new 1-dimensional distance based feature is created and used by the K-Nearest Neighbour classifier to classify each data sample into normal or abnormal. The performance of CANN was evaluated using accuracy, detection rate and true positive rate metrics on a public Cybersecurity dataset that contains traces of DDoS attacks. In [30], the authors proposed a two-stage machine learning architecture that uses (a) the K-Means clustering algorithm to detect an attack, and (b) Decision Trees (DT), Random Forest, Adaptive Boosting (AB) and Naive Bayes algorithms to classify several types of attacks. They evaluated their architecture on a public Cybersecurity dataset that includes traces of denial-of-service attacks, and showed that decision trees and random forest algorithms achieved the highest classification accuracy. In [31], the authors compared the performance of Modest Adaboost (MA), Real Adaboost (RA) and Gentle Adaboost (GA) on five public Cybersecurity and DDoS datasets. They showed that (a) the error rate of Modest Adaboost is higher compared to the error rates of Gentle and Real Adaboost and (b) Gentle and Real Adaboost have the same error rate performance. In [32], the authors proposed a method that uses L2 regularization and dropout techniques to improve the performance of Convolution Neural Networks (CNN) for IDS. They showed that their method achieved the highest precision, recall and F1-scores compared to several popular machine learning techniques, and evaluated their method on a Cybersecurity dataset that contains traces of DDoS attacks. In [33], the authors proposed a modified System Call Graph (SCG) that uses a Deep Neural Network (DNN) to integrate information from different detection techniques. They evaluated their approach on

three Cybersecurity datasets that include traces of DDoS attacks, and showed that their model achieved high detection rates and low false positives.

2.2 Correlation Analysis-based Intrusion Detection Systems

We focused on very recent works that developed Intrusion Detection Systems which integrated correlation techniques to detect DDoS attacks. In [34], the authors introduced a new feature selection method called *CorrCorr*. It uses the Multivariate Correlation (MC) and Addition-Based Correlation (ABC) methods to generate feature correlations and normal network traffic profiles from which anomalies that deviate from the normal profile are detected. They evaluated their method on two public Cybersecurity datasets that include traces of DDoS attacks. In [35], the authors present a tool that efficiently correlates cross-host attacks across multiple hosts. Their tool uses tagged provenance graphs that models the techniques and operational procedures used by an attacker. They define a novel Graph Similarity-based Alert Correlation (GSAC) technique that determines the entities that are associated with alerts generated on different hosts, and evaluated their tool on two public Cybersecurity datasets that contains attack traces on multiple hosts. In [36], the authors proposed a distributed denial-of-service attack detection method that combines the Enhanced Random Forest (ERF) ensemble learning method and an Optimized Genetic Algorithm (OGA). In [37], the authors demonstrated an approach that utilizes Multivariate Correlation analysis to identify DDoS attacks in real-time. In [38], the authors presented a correlation-based approach that transformed clusters of alerts into graph structures and computed signatures of repeated network patterns to characterize clusters of alerts. They evaluated their approach on real-world attack scenarios that include DDoS attacks. In [39], the authors proposed an efficient framework for correlating alerts in early warning systems. Their framework combines statistical and stream mining techniques to extract sequences of alerts that are part of multistep attack scenarios, and evaluated on two DDoS attack scenarios. In [40], the authors proposed a hybrid model that integrated Multi-Feature Correlation (MFC) and a deep neural network. They evaluated their model on the UNSW-NB15, AWID, CICIDS 2017 and CICIDS 2018 Cybersecurity datasets which include traces of DDoS attacks.

2.3 Summary

To the best of our knowledge, there is no work that compared multiple regression models to identify correlations of reflection attacks on the NTP servers and identify correlations of reflection attacks on the NetBIOS servers. A summary of the main attributes of the reviewed works is given in Table 1. Differently to the works in Table 1, we (a) developed an approach that evaluates the ability of the LASSO, Ridge and Elastic Net regression models in identifying correlations of reflection attacks on the NTP servers and correlations of reflection

attacks on the NetBIOS servers, (b) identify the devices and network traffic associated with the NTP and NetBIOS servers reflection attacks, and (c) identify the dwell times between reflection attacks on the NTP and NetBIOS servers.

Table 1 Summary of the main attributes of the reviewed works

Study	**Method**	**Dataset**	**Type of attack**
Singh Samra, R. et. al. [20]	NLP	IXP samples	Standard DDoS
Benmohamed, E. et. al.[21]	DNN	CICDDoS2017	Standard DDoS
Dasari, K.B. et. al. [22]	SVM	SNMP & DDoS	Reflection DDoS
Kshirsagar, D. et. al. [23]	IG & Correlation	CICDDoS2019, KDD Cup 1999	Reflection DDoS, Standard DDoS
Ahuja, V. et. al. [24]	J48 classifier	CICDDoS2019	Reflection DDoS
Najafimehr, M. et. al. [25]	DBSCAN, Euclidean	CICIDS2017, CICDDoS2019	Standard DDoS
Singh Samom, P. et. al. [26]	RF, NB, LR, KNN, MLP	CICDDoS2019, UNSW-NB15	Standard DDoS
Cil, A.E. et. al. [27]	Deep learning	CICDDoS2019	Standard DDoS
Hachmi, F. et. al. [28]	MOO, clustering, filtering	DARPA 1999, NSL-KDD, Env-data	Standard DDoS
Lin, W.-C. et. al. [29]	K-Means, KNN	KDD Cup 1999	Standard DDoS
Kaja, N. et. al. [30]	K-Means, DT, RF, AB, NB	KDD Cup 1999	Standard DDoS
Shahraki, A. et. al. [31]	MA, RA, GA	KDD Cup 1999, UNSW-NB15, TRAbID, CIC17, NSL-KDD,	Standard DDoS
Elsayed, M.S. et. al. [32]	CNN	InSDN	Standard DDoS
Mora-Gimeno, F.J. et. al. [33]	SCG, DNN	DARPA 1999, UNM, ADFA-LD	Standard DDoS
Gottwalt, F. et. al. [34]	MC, ABC	NSL-KDD, UNSW-NB15	Standard DDoS
Ghosh, S.K. et. al. [35]	GSAC	OpTC-NCR2, DARPA TC	Standard DDoS
Cheng, J. et. al. [36]	ERF, OGA	CAIDA 2007	Standard DDoS
More, K.K. et. al. [37]	MC	KDD Cup 1999, NSL-KDD, Custom	Standard DDoS
Haas, S. et. al. [38]	Graphs	DShield	Standard DDoS
Ramaki, A.A. et. al. [39]	Stream mining	DARPA 2000	Standard DDoS
Lei, S. et. al. [40]	MFC, DNN	CIC17, CIC18 UNSW, AWID	Standard DDoS
This paper	LASSO, Ridge, Elastic Net	NetFlow	Reflection DDoS

3 Network Model and Data

In this section, we present the network model to which our approach is applicable in Section 3.1. Then, we describe the NetFlow data in Section 3.2.

3.1 Network Model

Our approach is based on a generic client-server network model as depicted in Fig. 1 [41]. The network consists of client devices, servers and routers. Client devices are separate computers that access a service made available by a server. The server is another computer that the client accesses the service by way of the network. Traffic between these networks are managed by the router, which forwards packets to their destination Internet Protocol (IP) addresses. The workflow for an Intrusion Detection System (IDS) consists of two phases, as depicted in Fig. 2 [42]. In phase 1, network packet data between a client and a server, two clients or two servers are collected by the *Router*, and then the data is sent to the *Data store* which aggregates the data. Once the data is aggregated, in phase 2 the *Analysis console* retrieves the data and analyzes it to identify an attack.

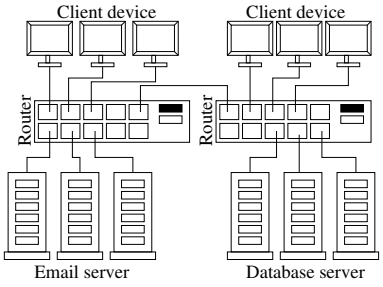


Fig. 1 Client-server network model

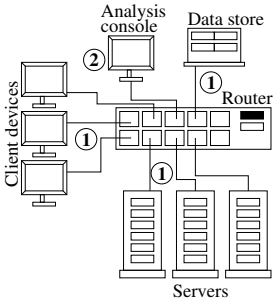


Fig. 2 An IDS setup

3.2 NetFlow Data

The NetFlow data is collected on most networks [43]. An example of a NetFlow record is given as follows:

```
6652800, 4666, Comp107130, Comp584379, 6, Port04167, 443, 130,
82, 71556, 55117
```

In this NetFlow record, there are 11 fields. The first field contains the start time (`6652800`). The second field contains the duration of the communications between the source and destination devices (`4666`). The third and fourth fields contain the source (`Comp107130`) and destination (`Comp584379`) devices, respectively. The fifth field contains the network protocol number (`6`, i.e., TCP). The sixth and seven fields contain the source (`Port04167`) and destination (`443`) ports, respectively. The eighth and ninth fields contain the

number of packets (130) and bytes (82) sent by the source device, respectively. The tenth and eleventh fields contain the number of packets (71556) and bytes (55117) sent by the destination device, respectively.

4 Identifying Correlations of Reflection Attacks

To execute a reflection attack, an attacker uses a tactic called IP (Internet Protocol) spoofing which replaces the real sender's source IP address with the IP address of another device, as depicted in Fig. 3. This causes the target device to respond to the request and send the answer to the victim host IP address. For example, a firewall may be configured to allow port 137 (i.e., NetBIOS) traffic so that computers on a local area network can communicate with network hardware and transmit data across the network. An attacker can take advantage of such a rule in the firewall and use some NetBIOS servers as intermediaries to execute a reflection attack on other NetBIOS servers.

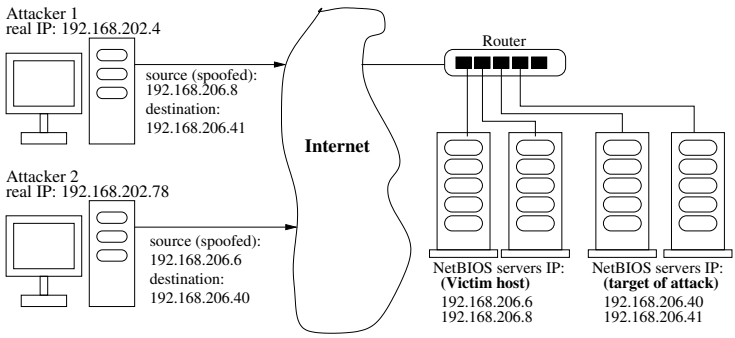


Fig. 3 IP address spoofing process.

Thus, our objective is to determine if reflection attacks are "correlated" or "not correlated". By "correlated", we mean the NetFlow records which are assigned the largest positive regression coefficients by the regression model. By "not correlated", we mean the NetFlow records which are assigned regression coefficients close to 0 by the regression model. In this paper, we aim to identify correlations of reflection attacks in the NetFlow data. The research problem that we address in this paper is given as follows: Given (a) the NetFlow data, (b) a network protocol number, and (c) a range of dates:

- Identify the NetFlow records which are assigned the largest positive regression coefficients or the smallest regression coefficients by the regression model.
- Identify the devices which are associated with the reflection attack and obtain the amount of traffic which is generated by the attack.
- Identify the time elapsed between the start times of two adjacent NetFlow records which are associated with the reflection attack.

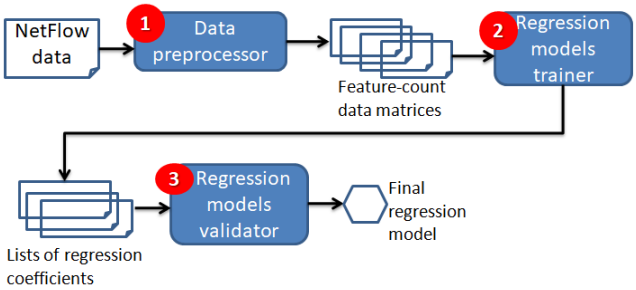


Fig. 4 Our approach consists of three modules: (a) Data preprocessor, (b) Regression models trainer, and (c) Regression models validator.

As such, the workflow we propose consists of three phases, as depicted in Fig. 4. The first phase in the workflow is *Data preprocessing*. It extracts the features in the NetFlow data and organizes the features into data structures. After the data structures are generated, the second phase of *Regression models training* applies different regression algorithms to learn the regression coefficients of multiple features given a target feature. This phase corresponds to "identifying" correlations of reflection attacks from the NetFlow data. Then, the third phase of *Regression models validation* applies statistical validation techniques to determine whether the regression model's estimated values are close to the observed values in the data. Next, we present the details for each of the three modules in the workflow.

4.1 Data Preprocessing

In the data preprocessing phase, the goal is to present the NetFlow data in a structured format so that the data can be easily processed by data analysis algorithms [44]. To attain this, we need to address three issues: (a) the NetFlow data contains vectors of network traffic, (b) the NetFlow records are unlabelled, and (c) the magnitude, range and unit of the feature values are different. By unlabelled, we mean that there are no NetFlow records labelled as "malicious" or "benign" in the NetFlow data.

4.1.1 Data Formatting

The NetFlow data is captured in a way such that the network traffic is represented by four vectors corresponding to all the NetFlow records for one day. The four vectors are: (a) the number of packets sent by the source device, (b) the number of bytes sent by the source device, (c) the number of packets sent by the destination device, and (d) the number of bytes sent by the destination device. To address this issue, we construct a feature-count data matrix. In this data matrix, the columns represent the NetFlow records and the rows represent the samples of a vector in the NetFlow records. To construct the feature-count data matrix, we implemented a function in the *Data preprocessor* module. The process is given as follows:

- Obtain the number of NetFlow records and initialize a feature-count data matrix, where the columns and rows of the data matrix are equal to the number of NetFlow records.
- Fill the diagonals in the data matrix with the vector value contained in number of packets sent by the source device, number of packets sent by the destination device, number of bytes sent by the source device or number of bytes sent by the destination device corresponding to the respective NetFlow record.
- Fill all the remaining cells in the data matrix with zero.

4.1.2 Data Scaling

In data scaling, the values in the data are transformed so that the values fit within a specific scale. The values in the NetFlow data vary in terms of the magnitude, range and unit. The size in the number of packets sent by a source device is typically lower than the size in the number of bytes sent by that source device. Furthermore, the range of values for the number of packets sent and the number of bytes sent are different. Thus, in order for a regression model to interpret the features on the same scale, we need to perform data scaling.

There are two standard methods for scaling data values [45]: (a) normalization, and (b) standardization. Data normalization scales the data values into a range of [0, n]. In contrast, data standardization scales the data values to have a mean of 0 and a standard deviation of 1. Data normalization is useful when the data is needed in the bounded intervals. However, it is difficult to identify an outlier. In contrast, data standardization produces useful information about outliers, which makes the regression model less sensitive to outliers [45]. Thus, we scale the values in the feature-count data matrix so that the values are centered around the mean with a unit standard deviation.

4.2 Regression Models Training

After the feature-count data matrix is generated, we need to identify (a) which NetFlow records are correlated during the reflection attack, and (b) which NetFlow records are not correlated during the reflection attack. To attain this, we use the Ridge, LASSO and Elastic Net regression models to obtain the regression coefficients for multiple NetFlow records given a target NetFlow.

Multiple linear regression (MLR) and polynomial regression (PLR) are standard regression algorithms that are widely used to model complex relationships with many variables [46]. Multiple linear regression models the relationship between the dependent variable and two or more independent variables using a straight line. In contrast, polynomial regression models the relationship between the dependent variable and two or more independent variables as a n^{th} degree polynomial. However, MLR and PLR models are susceptible to overfitting on the training data, which causes the model to perform poorly on new data. Differently to MLR and PLR models, which do not use regularization, the LASSO, Ridge and Elastic Net regression models use

regularization to constrain the regression coefficients and improve the model's accuracy. Regularization is achieved by penalizing variables that have a large coefficient value. The LASSO, Ridge and Elastic Net regression models functions are given in Table 2. The LASSO regression model includes a penalty term called $L1$-norm [47]. It sets the regression coefficients of some of the independent variables to zero. The Ridge regression model includes a penalty term called $L2$-norm. Differently to $L1$-norm, $L2$-norm shrinks the regression coefficients of all the independent variables towards zero [48]. The Elastic Net regression model includes both $L1$-norm and $L2$-norm penalty terms [49].

Table 2 Summary of Regularized Regression Models

Model	Penalty term	Objective
LASSO	$L1$-norm	Automatic feature selection
Ridge	$L2$-norm	Manual feature selection
Elastic Net	$L1$-norm + $L2$-norm	Semi-automatic feature selection

4.2.1 Handling Bias in Regularized Regression Models

To perform regression analysis, we need to address two issues: (a) handle bias in the regularized regression model, and (b) select the penalty parameter. In statistics, bias is anything that leads to a systematic difference between the observed values in the data and the estimates which are produced by a regression model [46]. The LASSO, Ridge and Elastic Net regression models add a penalty term in the cost function. The penalty term penalizes a regression model with large regression coefficients, which reduces the model's variance. For example, if the number of packets sent by NTP server A ranges from 80,000 to 120,000 per minute and the number of packets sent by NTP server B ranges from 800 to 1,200 per minute, the regression coefficient for the number of packets sent by NTP server B of 1 packet change will be a much larger coefficient in regard to its change in the number of packets sent compared to a 1 packet change in the number of packets sent by NTP server A. If a larger regression coefficient for NTP server B is obtained, then the regularized regression model will penalize NTP server B's regression coefficient. As a result, a biased model can be produced. To resolve this issue, we standardize all the values in the feature-count data matrix. Then, we input the standardized feature-count data matrix into the LASSO, Ridge and Elastic Net regression models, train the regression model and obtain the fitted regression model.

4.2.2 Selecting the penalty parameter

The penalty parameter (λ) is a value that controls the amount of shrinkage of the regression coefficients in the LASSO, Ridge and Elastic Net regression models [46]. When $\lambda = 0$, no regression coefficients are removed. When λ increases,

more regression coefficients are removed. When $\lambda = \infty$, all the regression coefficients are removed. To select the best value for λ, we use a general approach called *k-fold cross validation* [44]. It extracts a portion of the data and sets it aside to be used as a test set. The remaining portions of the data are used as the training set. The regression model is trained on the training dataset. Then, the test dataset is used to test the regression model. 10-fold cross-validation is typically used to obtain the best λ value [44]. We implemented a function in the *Regression models trainer* module to perform 10-fold cross validation and select the penalty parameter. The process is given in Algorithm 1:

Algorithm 1 Select the penalty parameter

Require: feature-count data matrix M, regression model P
  Initialize empty vector $lambda\_values[\,]$;
  $number\_lambdas \leftarrow 20$;
  $\lambda \leftarrow 0.1$;
  for $i \leftarrow 1$ to $number\_lambdas$ **do**
    $lambda\_values[i] \leftarrow \lambda$;
    $\lambda \leftarrow \lambda + 0.1$;
  end for
  Divide M into 10 folds;
  Initialize empty key-value pair vector $penalty\_mse[\,]$;
  for $j \leftarrow 1$ to 10 **do**
    Divide M_j into 10 parts;
    Assign parts 1 to 9 as the training set M_j^{Train};
    Assign part 10 as the test set M_j^{Test};
    for $k \leftarrow 1$ to $number\_lambdas$ **do**
      $\lambda \leftarrow lambda\_values[k]$;
      Assign λ to penalty parameter in regression model P;
      Train regression model P using M_j^{Train};
      Test regression model P using M_j^{Test};
      Obtain Mean Squared Error (MSE) from Test regression model P;
      $penalty\_mse[k] \leftarrow < \lambda, MSE >$;
    end for
  end for
  Obtain the λ associated with the smallest MSE from array $penalty\_mse$;

4.3 Regression Models Validation

Once the regression model is trained, we need to assess the model's accuracy. There are two standard metrics for measuring how close the values estimated by the regression model and the observed values in the data are. The metrics are [45]: (a) coefficient-of-determination (R^2), and (b) Root Mean Squared Error (RMSE). The coefficient-of-determination is the proportion of

variation in the dependent variable that is predictable from the independent variables. Differently to R^2, RMSE is the average difference between the regression model's estimated values and the observed values. A RMSE value ranges between 0 and infinity. If the RMSE value is close to 0, it shows that the regression model replicated the observed values accurately. However, it becomes difficult to interpret a large RMSE value. In contrast to the RMSE value, the R^2 value ranges between 0 and 1. If $R^2 = 0$, it shows that the regression model's estimated values are different from the observed values. If $R^2 = 1$, it shows that the regression model's estimated values match the observed values. Thus, we use the R^2 statistic to obtain the accuracy of the Ridge, LASSO and Elastic Net regression models.

4.3.1 Accounting for Inflation in R^2

The R^2 statistic is at least weakly increasing when more independent variables are added to the regression model. If redundant independent variables were included in the regression model, the R^2 value remains the same or increases. Consequently, the R^2 statistic alone cannot determine if the independent variables are useful. To resolve this issue, we obtain the adjusted R^2 value [50]. It determines whether adding more independent variables actually increases the regression model's fit. We implemented a function in the *Regression models validator* module to calculate the adjusted R^2. The formula for calculating the adjusted R^2 is [45]: $1 - \{(1 - R^2)(n - 1) \div (n - p - 1)\}$, where n is the number of NetFlow records associated with the reflection attack and p is the total number of independent variables.

5 Evaluation on an Enterprise Network

We conduct our study of reflection attacks on an enterprise network operated by Los Alamos National Laboratories. The network hosts 60,000 devices and provides storage and user account services. The NetFlow data is collected in the network [51]. One day's worth of NetFlow data contains 220,000,000 NetFlow records on average. All the NetFlow records are unlabeled. It was reported that the NetFlow data contains compromised devices [52], but the times and number of compromised devices are not known. Thus, we randomly select eight days worth of NetFlow data for analysis.

5.1 Phase 1: Identify Correlations of Reflection Attacks

To ascertain whether reflection attacks are correlated or not correlated, first we obtain the NetFlow records which are associated with a reflection attack. We implemented a function in our workflow to scan the NetFlow data and extract NetFlow records containing the same source and destination port numbers. We applied the function to the eight days of NetFlow data and identified reflection attacks on several network protocols, though we focused on a subset of attacks as reflection attacks on the NTP and NetBIOS servers. DDoS attacks on the

NTP and NetBIOS servers have been widely reported [12, 13]. For each day, we assigned the first NetFlow record as the dependent variable and assigned the remaining NetFlow records as independent variables. Then, we trained the Ridge, LASSO and Elastic Net regression models on the four attributes in the NetFlow data separately and obtained the fitted regression models. The four attributes are: (a) `number of packets sent by the source device`, (b) `number of bytes sent by the source device`, (c) `number of packets sent by the destination device`, and (d) `number of bytes sent by the destination device`.

5.1.1 Reflection Attack on the NTP Server

First, we obtain the R^2 and adjusted R^2 values for the Elastic Net, Ridge and LASSO regression models trained on the `number of packets sent by the source device` attribute. The adjusted R^2 shows if adding more NetFlow records in the LASSO, Ridge and Elastic Net regression models increases the R^2 value. To obtain the adjusted R^2 value, we set $p = 1$ and $n =$ the number of NetFlow records associated with the NTP server reflection attack. The R^2 and adjusted R^2 values are given in Table 3. From Table 3, we observed that (a) the R^2 and adjusted R^2 values for the Ridge regression model ranged from 0.01 to 0.03 on days 1, 3, 4, 5, 7 and 8, 0.05 on day 2 and 0.07 on day 6, (b) the R^2 and adjusted R^2 values for the Elastic Net regression model ranged from 0.01 to 0.03 on days 1, 4, 5, 6, 7 and 8, 0.04 on day 3 and 0.07 on day 2, and (c) the R^2 and adjusted R^2 values for the LASSO regression model ranged from 0.01 to 0.02 on days 1, 2, 3, 4, 6 and 7, 0.03 on day 8 and 0.06 on day 5. We obtained the R^2 and adjusted R^2 values for the Ridge, Elastic Net and LASSO regression models trained on the `number of bytes sent by the source device`, `number of packets sent by the destination device`, and `number of bytes sent by the destination device` attributes. Their R^2 and adjusted R^2 values ranged from 0.01 to 0.07 over the eight days.

On all the eight days, the R^2 and adjusted R^2 values for the Ridge, Elastic Net and LASSO regression models are close to 0, indicating that the accuracy of all three regression models are the same. Furthermore, the range of R^2 and adjusted R^2 values in all three regression models trained on the four attributes separately are the same. Moreover, the R^2 and adjusted R^2 values for the Ridge, LASSO and Elastic Net regression models are the same, indicating that more independent variables added to all the three regression models did not increase the regression model's fit to the observed data. Thus, the `number of packets sent by the source device` attribute can be used as the primary attribute.

Table 3 Accuracy of Ridge, Elastic Net (ENet) and LASSO regression models trained on the "*number of packets sent by the source device*" attribute

Day 1 ($n = 41125$)				**Day 2** ($n = 90891$)			
Metric	**Ridge**	**ENet**	**LASSO**	**Metric**	**Ridge**	**ENet**	**LASSO**
R^2	0.02	0.02	0.02	R^2	0.05	0.07	0.01
Adj. R^2	0.02	0.02	0.02	Adj. R^2	0.05	0.07	0.01
Day 3 ($n = 88800$)				**Day 4** ($n = 113804$)			
Metric	**Ridge**	**ENet**	**LASSO**	**Metric**	**Ridge**	**ENet**	**LASSO**
R^2	0.01	0.04	0.02	R^2	0.02	0.02	0.02
Adj. R^2	0.01	0.04	0.02	Adj. R^2	0.02	0.02	0.02
Day 5 ($n = 60863$)				**Day 6** ($n = 48336$)			
Metric	**Ridge**	**ENet**	**LASSO**	**Metric**	**Ridge**	**ENet**	**LASSO**
R^2	0.03	0.01	0.06	R^2	0.07	0.02	0.01
Adj. R^2	0.03	0.01	0.06	Adj. R^2	0.07	0.02	0.01
Day 7 ($n = 72081$)				**Day 8** ($n = 53942$)			
Metric	**Ridge**	**ENet**	**LASSO**	**Metric**	**Ridge**	**ENet**	**LASSO**
R^2	0.01	0.03	0.02	R^2	0.03	0.03	0.03
Adj. R^2	0.01	0.03	0.02	Adj. R^2	0.03	0.03	0.03

Next, we obtain the residuals from the Elastic Net, Ridge and LASSO regression models trained on the `number of packets sent by the source device` attribute. A residual is the difference between the regression model's estimated value and the observed value in the data. Residual analysis belongs to a class of techniques for evaluating the goodness-of-fit of a fitted regression model. If a regression model is a good fit to the observed data, all its residual values will be close to 0 or equals to 0. If a regression model is not a good fit to the observed data, some of its residual values will not be close to 0. To obtain the proportion of residuals, we implemented a function in the *Regression models validator* module. The process for obtaining the proportion of residuals is given as follows: (a) obtain the residual value for each sample in the regression model, (b) obtain the percentage of all unique residual values, and (c) obtain the cumulative distribution of the percentage of unique residual values. The proportion of residuals in the Elastic Net regression model for day 1 is shown in Fig. 5. From Fig. 5, we observed that (a) the residuals range from 0 to 80, and (b) a proportion of the residuals are greater than 0. When the residuals are greater than 0, it shows that the values estimated by the Elastic Net regression model differ from the observed values in the data. We obtained the proportion of residuals in the Elastic Net regression model for days 2 to 8. On all the 7 days, their residuals range from 0 to 80 and a proportion of those residuals are greater than 0. Next, we obtained the residuals from the Ridge and LASSO

regression models for days 1 to 8. On all the eight days, their residuals ranged from 0 to 80 and a proportion of those residuals are greater than 0.

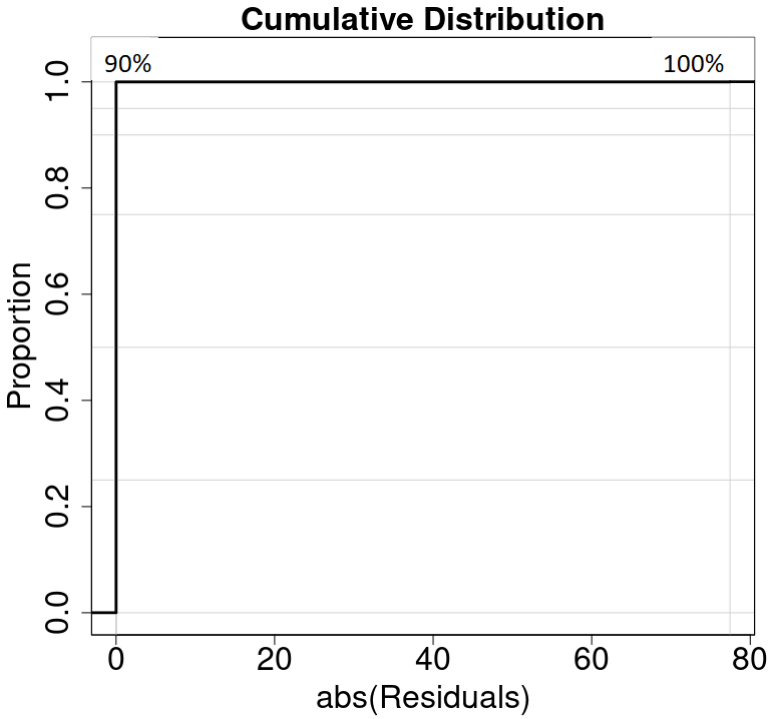


Fig. 5 Proportion of residuals in the Elastic Net regression model for day 1.

Next, we determine which one of three regression models best fit the data. To achieve this, we apply a standard technique called the general F-statistic. The F-statistic is used to compare statistical models that have been fitted to a dataset, in order to identify the statistical model that best describes the population from which the data is sampled [45]. First, we define the null and alternate hypotheses. The null hypothesis is that the sum-of-squares error (SSE) of one regression model is close to the SSE of a different regression model. The alternate hypothesis is that the SSE of one regression model differs significantly from the SSE of a different regression model. The formula for computing the general linear F-statistic is [45]: $(\frac{SSE(M_1)-SSE(M_2)}{df_{M1}-df_{M2}}) \div \frac{SSE(M_2)}{df_{M2}}$, where M_1 and M_2 are two different regression models, df_{M1} and df_{M2} are the degrees of freedom associated with regression models M_1 and M_2 respectively. When $F^* \geq 3.95$, we reject the null hypothesis in favour of the alternate hypothesis. We implemented a function in the *Regression model validator* module to obtain the F-statistic. We apply the general linear F-statistic on the Elastic Net, Ridge and LASSO regression models and obtained the F^* value.

A summary of F-tests on the Elastic Net, Ridge and LASSO regression models is given in Table 4. From Table 4, we observed that from days 1 to 8 the F^* value is 0. Since $F^* \leq 3.95$, we fail to reject the null hypothesis.

Table 4 F-test for the Elastic Net, Ridge and LASSO Regression Models

Day 1 ($n = 41125$)				**Day 2** ($n = 90891$)			
	SSE	D_f	F^*		**SSE**	D_f	F^*
Elastic Net	41124	1	**0**	Elastic Net	90890	1	**0**
LASSO	41124	1	-	LASSO	90890	1	-
Ridge	41124	1	**0**	Ridge	90890	1	**0**
LASSO	41124	1	-	LASSO	90890	1	-
Elastic Net	41124	1	**0**	Elastic Net	90890	1	**0**
Ridge	41124	1	-	Ridge	90890	1	-
Day 3 ($n = 88800$)				**Day 4** ($n = 113804$)			
	SSE	D_f	F^*		**SSE**	D_f	F^*
Elastic Net	88799	1	**0**	Elastic Net	113803	1	**0**
LASSO	88799	1	-	LASSO	113803	1	-
Ridge	88799	1	**0**	Ridge	113803	1	**0**
LASSO	88799	1	-	LASSO	113803	1	-
Elastic Net	88799	1	**0**	Elastic Net	113803	1	**0**
Ridge	88799	1	-	Ridge	113803	1	-
Day 5 ($n = 60863$)				**Day 6** ($n = 48336$)			
	SSE	D_f	F^*		**SSE**	D_f	F^*
Elastic Net	60862	1	**0**	Elastic Net	48335	1	**0**
LASSO	60862	1	-	LASSO	48335	1	-
Ridge	60862	1	**0**	Ridge	48335	1	**0**
LASSO	60862	1	-	LASSO	48335	1	-
Elastic Net	60862	1	**0**	Elastic Net	48335	1	**0**
Ridge	60862	1	-	Ridge	48335	1	-
Day 7 ($n = 72081$)				**Day 8** ($n = 53942$)			
	SSE	D_f	F^*		**SSE**	D_f	F^*
Elastic Net	72080	1	**0**	Elastic Net	53941	1	**0**
LASSO	72080	1	-	LASSO	53941	1	-
Ridge	72080	1	**0**	Ridge	53941	1	**0**
LASSO	72080	1	-	LASSO	53941	1	-
Elastic Net	72080	1	**0**	Elastic Net	53941	1	**0**
Ridge	72080	1	-	Ridge	53941	1	-

On all the eight days, a proportion of residuals in all three regression models are greater than 0, indicating that the values estimated by all three regression models differ from the observed values in the data. The residual values in all three regression models ranged from 0 to 80. Furthermore, the F^* value for all three regression models is 0. When (a) the F^* value is 0, (b) the range of residuals in all three regression models are the same, and (c) a proportion of residuals in all three regression models are greater than 0, the Elastic Net regression model can be used as the main model.

Next, we obtain the regression coefficients for all NetFlow records in the Elastic Net regression model. A summary of regression coefficients is given in Table 5. From Table 5, we observed that all regression coefficients obtained for days 1 to 8 are close to 0 or equal to 0. We obtained the regression coefficients of all NetFlow records from the Ridge and LASSO regression models for days 1 to 8. On all the eight days, the regression coefficients of all NetFlow records in the Ridge and LASSO regression models are close to 0 or equal to 0.

Table 5 Regression Coefficients in the Elastic Net Regression Model.

Day 1 ($n = 41125$)				**Day 2** ($n = 90891$)			
NetFlow records	4530	12032	24563	**NetFlow records**	6708	12947	71236
Coeff.	0.0004	0.001	0	**Coeff.**	0.0001	0.002	0
Day 3 ($n = 88800$)				**Day 4** ($n = 113804$)			
NetFlow records	7649	26196	54955	**NetFlow records**	47092	66712	-
Coeff.	0.001	0.005	0	**Coeff.**	0.003	0	-
Day 5 ($n = 60863$)				**Day 6** ($n = 48336$)			
NetFlow records	7407	9983	43473	**NetFlow records**	2968	23095	22273
Coeff.	0.0001	0.002	0	**Coeff.**	0.002	0.01	0
Day 7 ($n = 72081$)				**Day 8** ($n = 53942$)			
NetFlow records	12534	20617	38930	**NetFlow records**	10485	13798	29659
Coeff.	0.0001	0.001	0	**Coeff.**	0.001	0.03	0

On days 1 to 8, the regression coefficients of all NetFlow records associated with the NTP server reflection attack are close to 0 or equal to 0, indicating that reflection attacks on the NTP servers are not correlated.

5.1.2 Reflection Attack on the NetBIOS Server

As was done with the NTP servers, we obtain the R^2 and adjusted R^2 values for the Elastic Net, Ridge and LASSO regression models trained on the `number of packets sent by the source device` attribute. We set $p = 1$ and $n =$ the number of NetFlow records associated with the NetBIOS server reflection attack. The R^2 and adjusted R^2 values are given in Table 6. From Table 6, we observed that (a) the R^2 and adjusted R^2 values for the Ridge regression model ranged from 0.01 to 0.02 on days 1 to 8, (b) the R^2 and adjusted R^2 values for the Elastic Net regression model ranged from 0.01 to 0.02 on days 1 to 8, and (c) the R^2 and adjusted R^2 values for the LASSO regression model ranged from 0.01 to 0.02 on days 1 to 5, 7 and 8 and 0.04 on day 6. We obtained the R^2 and adjusted R^2 values from the Ridge, Elastic Net and LASSO regression models trained on the `number of bytes sent by the source device`, `number of packets sent by the destination device`, and `number of bytes sent by the destination device` attributes. Their R^2 and adjusted R^2 values ranged from 0.01 to 0.05 over the eight days.

On all the eight days, the R^2 and adjusted R^2 values from the Ridge, Elastic Net and LASSO regression models are close to 0, indicating that the accuracy of all the three regression models is the same. Furthermore, the range of R^2 and adjusted R^2 values from all the three regression models trained on the four attributes separately are the same. Moreover, the R^2 and adjusted R^2 values for the Ridge, LASSO and Elastic Net regression models are the same, indicating that more independent variables added to all the three regression models did not increase the regression model's fit to the observed data. Thus, the `number of packets sent by the source device` attribute can be used as the primary attribute.

Next, we obtain the residuals in the Elastic Net, Ridge and LASSO regression models trained on the `number of packets sent by the source device` attribute. The proportion of residuals in the Elastic Net regression model for day 1 is shown in Fig. 6. From Fig. 6, we observed that (a) the residuals range from 0 to 80, and (b) a proportion of the residuals are greater than 0. When the residuals are greater than 0, it shows that the values estimated by the Elastic Net regression model differ from the observed values in the data. We obtained the proportion of residuals in the Elastic Net regression model for days 2

Table 6 Accuracy of Ridge, Elastic Net (ENet) and LASSO regression models trained on the "*number of packets sent by the source device*" attribute

Day 1 ($n = 274017$)				**Day 2** ($n = 381342$)			
Metric	**Ridge**	**ENet**	**LASSO**	**Metric**	**Ridge**	**ENet**	**LASSO**
R^2	0.02	0.02	0.01	R^2	0.04	0.02	0.02
Adj. R^2	0.02	0.02	0.01	Adj. R^2	0.04	0.02	0.02
Day 3 ($n = 401589$)				**Day 4** ($n = 383427$)			
Metric	**Ridge**	**ENet**	**LASSO**	**Metric**	**Ridge**	**ENet**	**LASSO**
R^2	0.01	0.01	0.01	R^2	0.01	0.01	0.01
Adj. R^2	0.01	0.01	0.01	Adj. R^2	0.01	0.01	0.01
Day 5 ($n = 405236$)				**Day 6** ($n = 236899$)			
Metric	**Ridge**	**ENet**	**LASSO**	**Metric**	**Ridge**	**ENet**	**LASSO**
R^2	0.01	0.01	0.01	R^2	0.02	0.02	0.04
Adj. R^2	0.01	0.01	0.01	Adj. R^2	0.02	0.02	0.04
Day 7 ($n = 325218$)				**Day 8** ($n = 402723$)			
Metric	**Ridge**	**ENet**	**LASSO**	**Metric**	**Ridge**	**ENet**	**LASSO**
R^2	0.01	0.01	0.01	R^2	0.02	0.02	0.02
Adj. R^2	0.01	0.01	0.01	Adj. R^2	0.02	0.02	0.02

to 8. On all the 7 days, their residuals range from 0 to 80 and a proportion of those residuals are greater than 0. Next, we obtained the residuals in the Ridge and LASSO regression models for days 1 to 8. On all the eight days, the residuals in the Ridge and LASSO regression models ranged from 0 to 80 and a proportion of those residuals are greater than 0.

Next, we determine which one of three regression models best fit the data. We apply the general linear F-statistic on the Elastic Net, Ridge and LASSO regression models and obtained the F^* value. A summary of F-tests on the Elastic Net, Ridge and LASSO regression models is given in Table 7. From Table 7, we observed that from days 1 to 8 the F^* value is 0. Since $F^* \leq 3.95$, we fail to reject the null hypothesis.

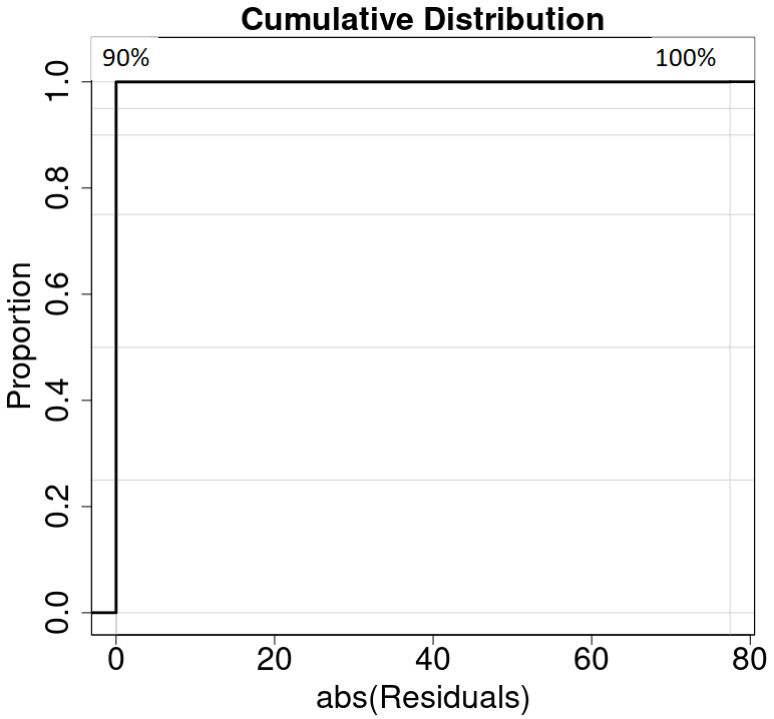


Fig. 6 Proportion of residuals in the Elastic Net regression model for day 1.

On all the eight days, a proportion of the residuals from the Elastic Net, Ridge and LASSO regression models are greater than 0, indicating that the values estimated by all the three regression models differ from the observed values in the data. The residuals in all the three regression models ranged from 0 to 80. Furthermore, the F^* value for the Elastic Net, Ridge and LASSO regression models is 0. When (a) the F^* value is 0, (b) the range of residuals in all the three regression models are the same, and (c) a proportion of residuals in all the three regression models are greater than 0, the Elastic Net regression model can be used as the main model.

Next, we obtain the regression coefficients for all NetFlow records in the Elastic Net regression model. A summary of the regression coefficients is given in Table 8. From Table 8, we observed that all the regression coefficients obtained for days 1 to 8 are close to 0 or equal to 0. We obtained the regression coefficients of all NetFlow records in the Ridge and LASSO regression models for days 1 to 8. On all the eight days, the regression coefficients of all NetFlow records in the Ridge and LASSO regression models are close to 0 or equal to 0.

Table 7 F-test for the Elastic Net, Ridge and LASSO Regression Models

Day 1 ($n = 274017$)				**Day 2** ($n = 381342$)			
	SSE	D_f	F^*		**SSE**	D_f	
Elastic Net	274016	1	**0**	Elastic Net	381341	1	**0**
LASSO	274016	1	-	LASSO	381341	1	-
Ridge	274016	1	**0**	Ridge	381341	1	**0**
LASSO	274016	1	-	LASSO	381341	1	-
Elastic Net	274016	1	**0**	Elastic Net	381341	1	**0**
Ridge	274016	1	-	Ridge	381341	1	-
Day 3 ($n = 401589$)				**Day 4** ($n = 383427$)			
	SSE	D_f	F^*		**SSE**	D_f	F^*
Elastic Net	401588	1	**0**	Elastic Net	383426	1	**0**
LASSO	401588	1	-	LASSO	383426	1	-
Ridge	401588	1	**0**	Ridge	383426	1	**0**
LASSO	401588	1	-	LASSO	383426	1	-
Elastic Net	401588	1	**0**	Elastic Net	383426	1	**0**
Ridge	401588	1	-	Ridge	383426	1	-
Day 5 ($n = 405236$)				**Day 6** ($n = 236899$)			
	SSE	D_f	F^*		**SSE**	D_f	F^*
Elastic Net	405235	1	**0**	Elastic Net	236898	1	**0**
LASSO	405235	1	-	LASSO	236898	1	-
Ridge	405235	1	**0**	Ridge	236898	1	**0**
LASSO	405235	1	-	LASSO	236898	1	-
Elastic Net	405235	1	**0**	Elastic Net	236898	1	**0**
Ridge	405235	1	-	Ridge	236898	1	-
Day 7 ($n = 325218$)				**Day 8** ($n = 402723$)			
	SSE	D_f	F^*		**SSE**	D_f	F^*
Elastic Net	325217	1	**0**	Elastic Net	402722	1	**0**
LASSO	325217	1	-	LASSO	402722	1	-
Ridge	325217	1	**0**	Ridge	402722	1	**0**
LASSO	325217	1	-	LASSO	402722	1	-
Elastic Net	325217	1	**0**	Elastic Net	402722	1	**0**
Ridge	325217	1	-	Ridge	402722	1	-

On days 1 to 8, the regression coefficients of all NetFlow records associated with the NetBIOS server reflection attack are close to 0 or equal to 0, indicating that the reflection attack on the NetBIOS servers are not correlated.

Table 8 Regression Coefficients in the Elastic Net Regression Model.

Day 1 ($n = 274017$)				**Day 2** ($n = 381342$)			
NetFlow records	102452	171565	-	**NetFlow records**	215831	165511	-
Coeff	0.0001	0	-	**Coeff**	0.0001	0	-
Day 3 ($n = 401589$)				**Day 4** ($n = 383427$)			
NetFlow records	29488	102419	269682	**NetFlow records**	192849	190578	-
Coeff	0.0002	0.00001	0	**Coeff**	0.001	0	-
Day 5 ($n = 405236$)				**Day 6** ($n = 236899$)			
NetFlow records	93279	311957	-	**NetFlow records**	46296	96846	93757
Coeff	0.001	0	-	**Coeff**	0.001	0.0001	0
Day 7 ($n = 325218$)				**Day 8** ($n = 402723$)			
NetFlow records	54203	271015	-	**NetFlow records**	43150	107872	251701
Coeff	0.0001	0	-	**Coeff**	0.001	0.001	0

5.2 Phase 2: Identify the Devices and Amount of Traffic Generated by the Reflection Attacks on NTP and NetBIOS Servers

The first phase of our analysis is characterized by the identification of correlations of reflection attacks on the NTP servers and correlations of reflection attacks on the NetBIOS servers. We observed that (a) reflection attacks on the NTP servers are not correlated, and (b) reflection attacks on the NetBIOS servers are not correlated. Our next objective is to identify the devices and the amount of traffic generated by the NTP servers and NetBIOS servers reflection attacks. To realize this, we obtain the source and destination devices which are associated with those reflection attacks.

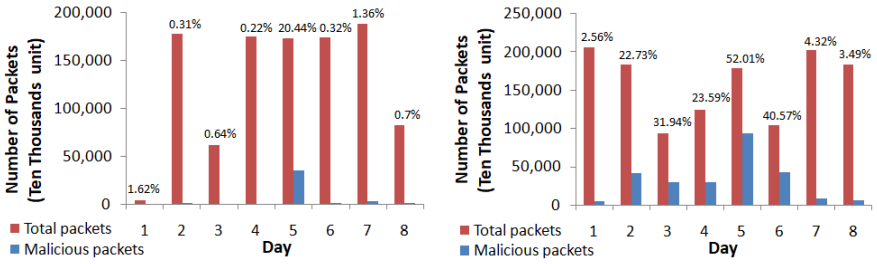
Therefore, we count the number of unique source and destination devices associated with the NTP servers reflection attacks. A summary of source and destination devices is given in Table 9. From Table 9, we observed that from day 1 to day 8, multiple source and destination devices are associated with reflection attacks on NTP servers.

Next, we obtain (a) the number of packets and bytes sent by these source and destination devices associated with these NTP servers reflection attacks, and (b) the total number of packets and bytes transmitted in the network. The total number of packets, number of malicious packets and percentage of malicious packets are shown in Fig. 7. From Fig. 7(a), we observed that the percentage of malicious packets sent by these source devices range from 0.22% to 20.44% over eight days. From Fig. 7(b), we observed that the percentage

Table 9 Devices Associated with NTP Server Reflection Attacks

Day	Qty. source devices	Qty. destination devices	Source packets	Destination packets
1	154	85	682,853	52,715,588
2	230	538	5,451,544	417,472,580
3	208	663	3,914,066	300,596,636
4	247	623	3,807,407	292,454,884
5	227	555	354,300,063	930,826,308
6	381	500	5,517,178	422,559,364
7	188	541	25,678,529	87,543,748
8	124	492	5,793,229	64,289,836

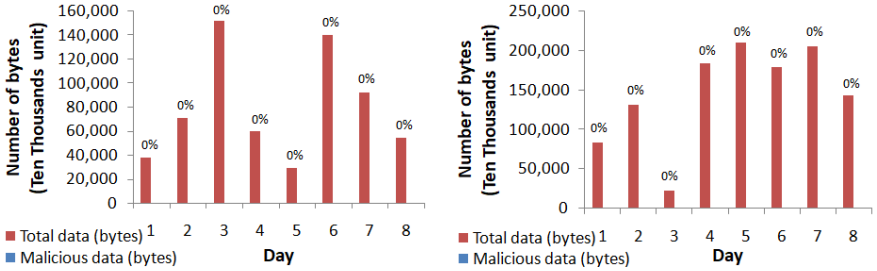
of malicious packets sent by these destination devices range from 2.56% to 52.01% over eight days. The total number of bytes, number of bytes contained in the malicious packets and percentage of bytes in those malicious packets are shown in Fig. 8. From Fig. 8(a), we observed that the percentage of bytes in those malicious packets sent by these source devices is 0% on all eight days. From Fig. 8(b), we observed that the percentage of bytes in the malicious packets sent by these destination devices is 0% on all eight days. This result shows that the malicious packets associated with the reflection attack on these NTP servers contained 0-byte payloads.



(a) Source devices. (b) Destination devices.

Fig. 7 Total number of packets, number of malicious packets and percentage of malicious packets transmitted in the network.

While the percentage of malicious packets sent by these source devices ranged from 0.22% to 20.44% and the percentage of malicious packets sent by these destination devices ranged from 2.56% to 52.01% over eight days, all the malicious packets contained 0-byte payloads, indicating that the reflection attack did not overwhelm these NTP servers.



(a) Source devices.

(b) Destination devices.

Fig. 8 Total number of bytes, number of bytes contained in the malicious packets and percentage of bytes in those malicious packets transmitted in the network.

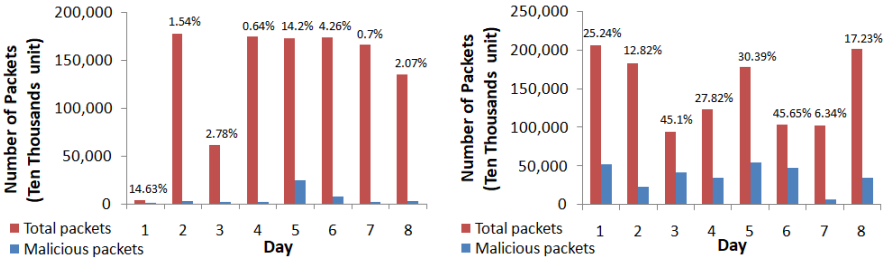
As was done with the NTP servers, we count the number of unique source and destination devices which are associated with the NetBIOS server reflection attack. A summary of source and destination devices is given in Table 10. From Table 10, we observed that from day 1 to day 8, multiple source and destination devices are associated with reflection attacks on NetBIOS servers.

Table 10 Devices Associated with NetBIOS Server Reflection Attacks

Day	Qty. source devices	Qty. destination devices	Source packets	Destination packets
1	57	43	6,171,080	520,546,040
2	69	81	27,483,909	235,479,535
3	67	114	17,089,572	424,362,101
4	57	57	11,205,241	344,953,694
5	50	108	246,062,878	543,844,657
6	540	95	74,180,774	475,430,314
7	275	205	11,666,399	65,478,293
8	78	327	28,006,385	348,190,008

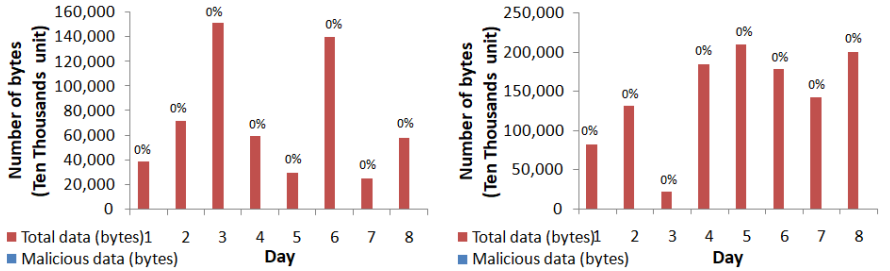
Next, we obtain (a) the number of packets and bytes sent by these source and destination devices associated with the reflection attack on these NetBIOS servers, and (b) the total number of packets and bytes transmitted in the network. The total number of packets, number of malicious packets and percentage of malicious packets are shown in Fig. 9. From Fig. 9(a), we observed that the percentage of malicious packets sent by these source devices range from 0.64% to 14.63% over eight days. From Fig. 9(b), we observed that the percentage of malicious packets sent by these destination devices range from 6.34% to 45.65% over eight days. The total number of bytes, number of bytes contained in the malicious packets and the percentage of bytes in those malicious packets are shown in Fig. 10. From Fig. 10(a), we observed that the percentage of bytes in those malicious packets sent by these source devices is 0% on all eight days. From Fig. 10(b), we observed that the percentage of

bytes in those malicious packets sent by these destination devices is 0% on all eight days. This result shows that the malicious packets associated with the reflection attack on these NetBIOS servers contained 0-byte payloads.



(a) Source devices. (b) Destination devices.

Fig. 9 Total number of packets, number of malicious packets and percentage of malicious packets transmitted in the network.



(a) Source devices. (b) Destination devices.

Fig. 10 Total number of bytes, number of bytes contained in the malicious packets and percentage of bytes in those malicious packets transmitted in the network.

While the percentage of malicious packets sent by these source devices ranged from 0.64% to 14.63% and the percentage of malicious packets sent by these destination devices ranged from 6.34% to 45.65% over eight days, those malicious packets contained a 0-byte payload, indicating that the reflection attack did not overwhelm these NetBIOS servers.

5.3 Phase 3: Identify the Dwell Times of Reflection Attacks on NTP and NetBIOS Servers

The second phase of our analysis is characterized by the identification of devices associated with the NTP and NetBIOS server reflection attacks and

the network traffic generated by those attacks. We observed that (a) multiple source and destination devices are associated with those reflection attacks, and (b) a small percentage of network traffic is generated by the NTP and NetBIOS server reflection attacks. Our next objective is to identify the dwell time of NTP and NetBIOS server reflection attacks. To achieve this, we obtain the time elapsed between the start times of adjacent NetFlow records associated with the reflection attack.

The dwell time for reflection attacks on the NTP server on days 1 to 8 are shown in Fig. 11. From Fig. 11(a) to Fig. 11(h), we observed that the dwell time ranged from 0 seconds to 68 seconds over eight days.

> The dwell times of NTP server reflection attacks ranged from 0 seconds to 68 seconds over eight days, indicating that the time elapsed between reflection attacks on these NTP servers are small.

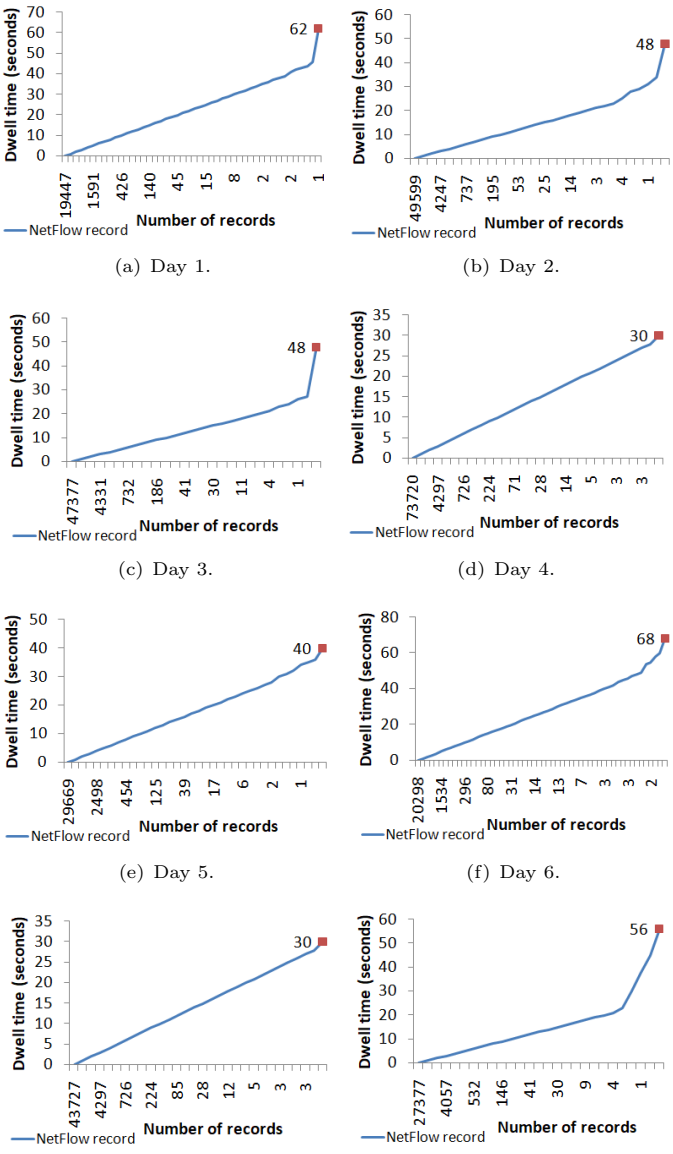
As was done with the NTP servers, we obtain the dwell time for reflection attacks on the NetBIOS server. The dwell time on days 1 to 8 are shown in Fig. 12. From Fig. 12(a) to Fig. 12(h), we observed that the dwell time ranged from 0 seconds to 198 seconds over eight days.

> The dwell time of NetBIOS server reflection attacks ranged from 0 seconds to 198 seconds over eight days, indicating that the time elapsed between reflection attacks on these NetBIOS servers are small.

6 Discussion

From these results, we have shown that the LASSO, Ridge and Elastic Net regression models are unsuitable as a means for identifying correlations of reflection attacks. Our analysis of the NetFlow data from a large enterprise network helps to bring awareness to the extent to which reflection attacks are correlated. The fact that reflection attacks on these NTP servers are not correlated and reflection attacks on these NetBIOS servers are not correlated is not obvious. For example, the regression coefficients in the Elastic Net, LASSO and Ridge regression models from day 1 to day 8 are close to 0 or equal to 0. We summarize our findings and recommendations in Table 11.

We observed that the network traffic generated by these reflection attacks did not overwhelm the NTP and NetBIOS server on all eight days. While network administrators are less concerned with a reflection attack that did not lead to a loss of network service, it is better to equip the network with reflection attack detectors to reduce the network service downtime. These recommendations are suitable for various networks as well, since complex networks including but not limited to peer-to-peer networks and Internet-of-Things networks can also benefit from NetFlow data analysis.



(a) Day 1. (b) Day 2.

(c) Day 3. (d) Day 4.

(e) Day 5. (f) Day 6.

(g) Day 7. (h) Day 8.

Fig. 11 Dwell times of NTP server reflection attacks.

7 Threats to Validity

We have identified the following threats to validity: (a) internal validity and (b) external validity.

Internal validity is concerned with the extent to which a cause-and-effect relationship established in an empirical study cannot be explained by other

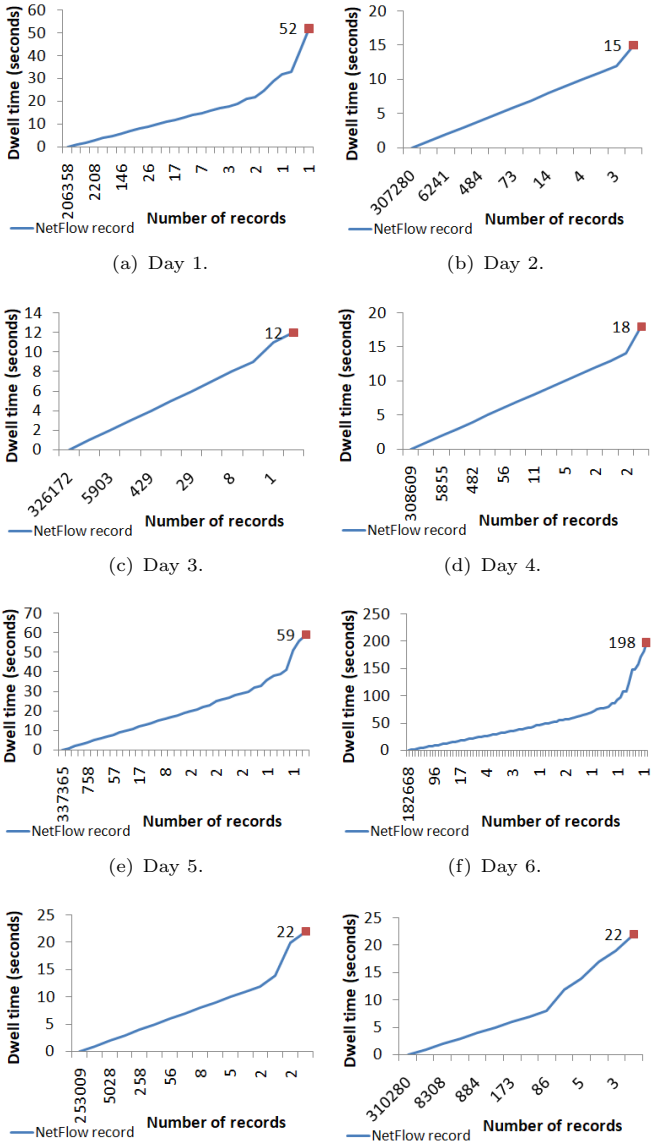


Fig. 12 Dwell times of NetBIOS server reflection attacks.

factors [53]. Those factors are comprised of: (a) the quality of the NetFlow data which can lead to variations in the network traffic over time, and thus it could mislead our correlation analysis, (b) the choice of dates of NetFlow data which can lead to selection bias, and (c) the types of data analyzed in our study. Regarding the quality of the NetFlow data, M.M. Turcotte *et al.* [51] showed that the volume of network packets captured in the NetFlow data

Table 11 Findings and recommendations.

Finding	Recommendation
Reflection attacks on the NTP and NetBIOS servers are not correlated in the NetFlow data.	Small percentages of network packets can be ignored unless a large percentage of network packets associated with a reflection attack is observed in the NetFlow data.
Spoofed requests triggered reflection attacks on the NTP and NetBIOS servers	Network administrators can implement ingress filtering on their networks which allows detection of IP packet spoofing.
The dwell times between reflection attacks on the NTP and NetBIOS servers are short.	Network attack mitigation schemes can implement Anycast to scatter the attack traffic and absorb the attack.
The LASSO, Ridge and Elastic Net models did not identify correlations of reflection attacks.	Conducting an empirical study of various deep-learning techniques could improve detection of a reflection attack.

matched the expected traffic in the enterprise network operated by Los Alamos National Laboratories, which assured that the NetFlow data is high quality. With respect to the dates of NetFlow data, we may have missed reflection attacks which are not correlated. To address this issue, we randomly selected eight days worth of NetFlow data for our analysis. On the types of data analyzed in our study, we did not consider the Windows server logs [54], hardware performance data [55] or behaviour logs [56] as they are beyond the scope of this paper, nor perform deep packet inspection because it would require substantial resources out of the reach of this paper. Having said that, we showed that reflection attacks on the NTP and NetBIOS servers exist in the NetFlow data and those reflection attacks are not correlated.

External validity is concerned with the extent to which the results presented in an empirical study can be generalized to other settings [53]. Some data centers do not collect detailed security incident reports, and some may not release the security logs due to privacy concerns [57]. Our conclusions are based on the NetFlow data of a large enterprise network, and the results we presented in our study may not generalize to other network models. Consequently, this makes our statistical analysis difficult to confirm. Having said that, network monitoring tools are currently being deployed on these networks [58]. Therefore, validating our analysis has become attainable.

8 Conclusion and Future Work

An approach based on correlating NetFlow records in the NetFlow data is proposed to identify correlations of reflection attacks. We showed that reflection attacks on the NTP and NetBIOS servers exist in the NetFlow data and evaluated the Ridge, Elastic Net and LASSO regression models. We applied the k-fold cross validation and coefficient-of-determination and ensured accurate results. From our study, we learned that (a) reflection attacks on the NTP servers are not correlated, (b) reflection attacks on the NetBIOS servers

are not correlated, (c) the dwell time between reflection attacks on the NTP and NetBIOS server is short, and (d) a small percentage of network traffic is generated by reflection attacks on the NTP and NetBIOS server.

In our future work, we plan to apply our approach on NetFlow data from more networks, and identify correlations of reflection attacks other than reflection attacks on the NTP and NetBIOS servers.

Acknowledgements

This work is supported by the Engineering and Physical Sciences Research Council [Grant number: EP/V026763/1]. We would like to thank the anonymous reviewers for their constructive feedback, which helped improve our paper significantly.

Declarations

Ethical Approval

Not applicable

Competing interests

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Authors' contributions

Edward Chuah prepared the manuscript and conducted the experiments. Neeraj Suri reviewed and edited the manuscript.

Availability of data and materials

The NetFlow data analyzed during this study is available for downloading at https://csr.lanl.gov/data/2017/. The codes used in this study are available from the corresponding author upon request.

References

[1] Chordiya, A.R., Majumder, S., Javaid, A.Y.: Man-in-the-middle (MITM) attack based hijacking of HTTP traffic using open source tools. In: Proceedings of IEEE International Conference on Electro/Information Technology (EIT), pp. 0438–0443 (2018). https://doi.org/10.1109/EIT.2018.8500144

[2] Bian, H., Bai, T., Salahuddin, M.A., Limam, N., Daya, A.A., Boutaba, R.: Host in danger? detecting network intrusions from authentication

logs. In: Proceedings of 15th International Conference on Network and Service Management (CNSM), pp. 1–9 (2019). https://doi.org/10.23919/CNSM46954.2019.9012700

[3] Friedberg, I., Skopik, F., Settanni, G., Fiedler, R.: Combating advanced persistent threats: From network event correlation to incident detection. Computers and Security **48**, 35–57 (2015) https://doi.org/10.1016/j.cose.2014.09.006

[4] Noble, J., Adams, N.M.: Correlation-based streaming anomaly detection in cyber-security. In: Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW), pp. 311–318 (2016). https://doi.org/10.1109/ICDMW.2016.0051

[5] Shin, M.S., Jeong, K.J.: Alert correlation analysis in intrusion detection. In: International Conference on Advanced Data Mining and Applications, pp. 1049–1056 (2006). https://doi.org/10.1007/11811305_114 . Springer

[6] Haas, S., Fischer, M.: On the alert correlation process for the detection of multi-step attacks and a graph-based realization. ACM SIGAPP Applied Computing Review **19**(1) (2019) https://doi.org/10.1145/3325061.3325062

[7] Cheng, Q., Wu, C., Zhou, S.: Discovering attack scenarios via intrusion alert correlation using graph convolutional networks. IEEE Communications Letters **25**(5), 1564–1567 (2021) https://doi.org/10.1109/LCOMM.2020.3048995

[8] Negi, C.S., Kumari, N., Kumar, P., Sinha, S.K.: An approach for alert correlation using ArcSight SIEM and open source NIDS. In: Nath, V., Mandal, J.K. (eds.) Proceeding of Fifth International Conference on Microelectronics, Computing and Communication Systems, pp. 29–40. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-0275-7_3

[9] Zadnik, M., Wrona, J., Hynek, K., Cejka, T., Husák, M.: Discovering coordinated groups of IP addresses through temporal correlation of alerts. IEEE Access **10**, 82799–82813 (2022) https://doi.org/10.1109/ACCESS.2022.3196362

[10] Chuah, E., Suri, N., Jhumka, A., Alt, S.: Challenges in identifying network attacks using netflow data. In: Proceedings of IEEE International Symposium on Network Computing and Applications (NCA) (2021). https://doi.org/10.1109/NCA53618.2021.9685305

[11] Gondim, J.J.C., de Oliveira Albuquerque, R., Sandoval Orozco, A.L.: Mirror saturation in amplified reflection distributed denial of service: A case

of study using SNMP, SSDP, NTP and DNS protocols. Future Generation Computer Systems **108**, 68–81 (2020) https://doi.org/10.1016/j.future.2020.01.024

[12] Sarmento, A.G., Yeo, K.C., Azam, S., Karim, A., Al Mamun, A., Shanmugam, B.: Applying big data analytics in DDos forensics: Challenges and opportunities. In: Jahankhani, H., Jamal, A., Lawson, S. (eds.) Cybersecurity, Privacy and Freedom Protection in the Connected World, pp. 235–252. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68534-8_15

[13] Kopp, D., Dietzel, C., Hohlfeld, O.: DDoS never dies? An IXP perspective on DDoS amplification attacks. In: Passive and Active Measurement, pp. 284–301. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72582-2_17

[14] Anagnostopoulos, M., Lagos, S., Kambourakis, G.: Large-scale empirical evaluation of DNS and SSDP amplification attacks. Journal of Information Security and Applications **66**, 103168 (2022) https://doi.org/10.1016/j.jisa.2022.103168

[15] Joshi, J.: Network Security: Know It All, p. 368. Morgan Kaufmann, USA (2008)

[16] Liu, Z., Jin, H., Hu, Y., Bailey, M.: Practical proactive DDoS-attack mitigation via endpoint-driven in-network traffic control. IEEE/ACM Transactions on Networking **26**(4), 1948–1961 (2018) https://doi.org/10.1109/TNET.2018.2854795

[17] Chawla, S., Sachdeva, M., Behal, S.: Discrimination of DDoS attacks and flash events using pearson's product moment correlation method. International Journal of Computer Science and Information Security (IJCSIS) **14**(10) (2016)

[18] Hostiadi, D.P., Ahmad, T.: Hybrid model for bot group activity detection using similarity and correlation approaches based on network traffic flows analysis. Journal of King Saud University - Computer and Information Sciences **34**(7), 4219–4232 (2022) https://doi.org/10.1016/j.jksuci.2022.05.004

[19] Heryanto, A., Stiawan, D., Bin Idris, M.Y., Bahari, M.R., Hafizin, A.A., Budiarto, R.: Cyberattack feature selection using correlation-based feature selection method in an intrusion detection system. In: Proceedings of the International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), pp. 79–85 (2022). https://doi.org/10.23919/EECSI56542.2022.9946449

[20] Singh Samra, R., Barcellos, M.: Ddos2vec: Flow-level characterisation of volumetric DDoS attacks at scale. Proceedings of the ACM on Networking **1**(CoNEXT3), 1–25 (2023) https://doi.org/10.1145/3629135

[21] Benmohamed, E., Thaljaoui, A., El Khediri, S., Aladhadh, S., Alohali, M.: DDoS attacks detection with half autoencoder-stacked deep neural network. International Journal of Cooperative Information Systems, 2350025 (2023) https://doi.org/10.1142/S0218843023500259

[22] Dasari, K.B., Devarakonda, N.: Evaluation of svm kernels with multiple uncorrelated feature subsets selected by multiple correlation methods for reflection amplification DDoS attacks detection, 99–111 (2023) https://doi.org/10.1007/978-981-19-6791-7_6

[23] Kshirsagar, D., Kumar, S.: A feature reduction based reflected and exploited DDoS attacks detection system. Journal of Ambient Intelligence and Humanized Computing, 1–13 (2022) https://doi.org/10.1007/s12652-021-02907-5

[24] Ahuja, V., Kotkar, M., Bhongade, R., Kshirsagar, D.: Reflection based distributed denial of service attack detection system. In: Proceedings of the 6th IEEE International Conference On Computing, Communication, Control And Automation (ICCUBEA), pp. 1–5 (2022). https://doi.org/10.1109/ICCUBEA54992.2022.10011055 . IEEE

[25] Najafimehr, M., Zarifzadeh, S., Mostafavi, S.: A hybrid machine learning approach for detecting unprecedented DDoS attacks. The Journal of Supercomputing **78**(6), 8106–8136 (2022) https://doi.org/10.1007/s11227-021-04253-x

[26] Singh Samom, P., Taggu, A.: Distributed denial of service (DDoS) attacks detection: A machine learning approach. In: Applied Soft Computing and Communication Networks: Proceedings of ACN 2020, pp. 75–87 (2021). https://doi.org/10.1007/978-981-33-6173-7_6 . Springer

[27] Cil, A.E., Yildiz, K., Buldu, A.: Detection of DDoS attacks with feed forward based deep neural network model. Expert Systems with Applications **169**, 114520 (2021) https://doi.org/10.1016/j.eswa.2020.114520

[28] Hachmi, F., Boujenfa, K., Limam, M.: Enhancing the accuracy of intrusion detection systems by reducing the rates of false positives and false negatives through multi-objective optimization. Journal of Network and System Management **27**(1), 93–120 (2019) https://doi.org/10.1007/s10922-018-9459-y

[29] Lin, W.-C., Ke, S.-W., Tsai, C.-F.: CANN: An intrusion detection system

based on combining cluster centers and nearest neighbors. Knowledge-Based Systems **78**, 13–21 (2015) https://doi.org/10.1016/j.knosys.2015.01.009

[30] Kaja, N., Shaout, A., Ma, D.: An intelligent intrusion detection system. Applied Intelligence **49**(9), 3235–3247 (2019) https://doi.org/10.1007/s10489-019-01436-1

[31] Shahraki, A., Abbasi, M., Haugen: Boosting algorithms for network intrusion detection: A comparative evaluation of real adaboost, gentle adaboost and modest adaboost. Engineering Applications of Artificial Intelligence **94** (2020) https://doi.org/10.1016/j.engappai.2020.103770

[32] Elsayed, M.S., Jahromi, H.Z., Nazir, M.M., Jurcut, A.D.: The role of CNN for intrusion detection systems: an improved CNN learning approach for SDNs. In: Proceedings of the International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures, pp. 91–104 (2021). https://doi.org/10.1007/978-3-030-78459-1_7 . Springer

[33] Mora-Gimeno, F.J., Mora-Mora, H., Volckaert, B., Atrey, A.: Intrusion detection system based on integrated system calls graph and neural networks. IEEE Access **9**, 9822–9833 (2021) https://doi.org/10.1109/ACCESS.2021.3049249

[34] Gottwalt, F., Chang, E., Dillon, T.: CorrCorr: A feature selection method for multivariate correlation network anomaly detection techniques. Computers & Security **83**, 234–245 (2019) https://doi.org/10.1016/j.cose.2019.02.008

[35] Ghosh, S.K., Satvat, K., Gjomemo, R., Venkatakrishnan, V.N.: Ostinato: Cross-host attack correlation through attack activity similarity detection. In: Badarla, V.R., Nepal, S., Shyamasundar, R.K. (eds.) Information Systems Security, pp. 1–22. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-23690-7_1

[36] Cheng, J., Li, M., Tang, X., Sheng, V.S., Liu, Y., Guo, W.: Flow correlation degree optimization driven random forest for detecting DDoS attacks in cloud computing. Security and Communication Networks (2018) https://doi.org/10.1155/2018/6459326

[37] More, K.K., Gosavi, P.B.: A real time system for denial of service attack detection based on multivariate correlation analysis approach. In: Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pp. 1125–1131 (2016). https://doi.org/10.1109/ICEEOT.2016.7754860

[38] Haas, S., Wilkens, F., Fischer, M.: Efficient attack correlation and identification of attack scenarios based on network-motifs. In: Proceedings of the IEEE International Performance Computing and Communications Conference (IPCCC), pp. 1–11 (2019). https://doi.org/10.1109/IPCCC47392.2019.8958734

[39] Ramaki, A.A., Amini, M., Ebrahimi Atani, R.: RTECA: Real time episode correlation algorithm for multi-step attack scenarios detection. Computers & Security **49**, 206–219 (2015) https://doi.org/10.1016/j.cose.2014.10.006

[40] Lei, S., Xia, C., Li, Z., Li, X., Wang, T.: HNN: A novel model to study the intrusion detection based on multi-feature correlation and temporal-spatial analysis. IEEE Transactions on Network Science and Engineering **8**(4), 3257–3274 (2021) https://doi.org/10.1109/TNSE.2021.3109644

[41] Halsall, F.: Data Communications, Computer Networks and Open Systems. Addison-Wesley, USA (1996)

[42] Mancini, L.V., Pietro, R.: Intrusion Detection Systems, p. 250. Springer, USA (2008)

[43] Lin, H., Yan, Z., Chen, Y., Zhang, L.: A survey on network security-related data collection technologies. IEEE Access **6**, 18345–18365 (2018) https://doi.org/10.1109/ACCESS.2018.2817921

[44] Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, USA (2006)

[45] Agresti, A., Franklin, C.: Statistics: The Art and Science of Learning From Data. Prentice Hall, USA (2009)

[46] Schroeder, L.D., Sjoquist, D.L., Stephan, P.E.: Understanding Regression Analysis: An Introductory Guide vol. 57. Sage Publications, USA (2016)

[47] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) **58**(1), 267–288 (1996) https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

[48] Hoerl, A.E., Kennard, R.W.: Ridge regression: Biased estimation for nonorthogonal problems. Technometrics **42**(1), 80–86 (2000) https://doi.org/10.1080/00401706.1970.10488634

[49] Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society. Series B (Statistical Methodology) **67**(2), 301–320 (2005) https://doi.org/10.1111/j.1467-9868.2005.00503.x

[50] Yin, P., Fan, X.: Estimating R^2 shrinkage in multiple regression: A comparison of different analytical methods. The Journal of Experimental Education **69**(2), 203–224 (2001) https://doi.org/10.1080/00220970109600656

[51] Turcotte, M.M., Kent, A.D., Hash, C.: Unified host and network data set. Data Science for Cyber Security, 16 (2018) https://doi.org/10.1142/9781786345646_001

[52] Zhenzheng, H., He, Q., Chuah, E., al.: Developing data science tools for improving enterprise cyber-security. In: The Alan Turing Institute Data Study Group Final Report (2018). https://doi.org/10.5281/zenodo.3558251

[53] Slack, M.K., Draugalis, J. Jolaine R.: Establishing the internal and external validity of experimental studies. American Journal of Health-System Pharmacy **58**(22), 2173–2181 (2001) https://doi.org/10.1093/ajhp/58.22.2173

[54] Dwyer, J., Truta, T.M.: Finding anomalies in windows event logs using standard deviation. In: Proceedings of the IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp. 563–570 (2013). https://doi.org/10.4108/icst.collaboratecom.2013.254136

[55] Yu, M., Halak, B., Zwolinski, M.: Using hardware performance counters to detect control hijacking attacks. In: Proceedings of the IEEE International Verification and Security Workshop (IVSW) (2019). https://doi.org/10.1109/IVSW.2019.8854399

[56] Shalaginov, A., Franke, K., Huang, X.: Malware beaconing detection by mining large-scale DNS logs for targeted attack identification. International Journal of Computer and Systems Engineering **10**(4), 743–755 (2016) https://doi.org/10.5281/zenodo.1123927

[57] Miranskyy, A., Hamou-Lhadj, A., Cialini, E., Larsson, A.: Operational-log analysis for big data systems: Challenges and solutions. IEEE Software **33**(2), 52–59 (2016) https://doi.org/10.1109/MS.2016.33

[58] Ghafir, I., Prenosil, V., Svoboda, J., Hammoudeh, M.: A survey on network security monitoring systems. In: Proceedings of the IEEE International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), pp. 77–82 (2016). https://doi.org/10.1109/W-FiCloud.2016.30