Defense against Bandwidth Attacks with Traffic Resource Management

Hongli Luo
Department of Electrical and Computer Engineering Technology
Indiana University - Purdue University Fort Wayne
Fort Wayne, IN 46805, USA
luoh@ipfw.edu

Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL 33124, USA
shyu@miami.edu

Abstract

In this paper, a framework is proposed to defend against Internet bandwidth attacks with traffic resource management to provide service for legitimate users. Denial of Service (DoS) is one of the major bandwidth attacks in the Internet. A DoS attack generates a large volume of traffic to consume the network bandwidth and degrade the service that legitimate users can obtain. Incoming traffic to the server is monitored and features are extracted for each connection. Anomaly detection technique is used to detect the abnormal traffic. Based on the outcome of the anomaly detection technique, the proposed resource management approach allocates suitable bandwidth. With the early detection of DoS, the attack traffic can be isolated. The bandwidth occupied by the attack can be reduced and protected for the legitimate users. Performances are compared under different attack loads with and without resource management. Simulation results show that bandwidth can be greatly saved from an attack and the service for the legitimate users can be protected during an attack.

1 Introduction

DoS is a major threat to network security and the availability of Internet services. The goal of DoS attacks is to completely occupy the resources of a server, which prevents legitimate users from accessing the service. DoS attacks can also deplete the link bandwidth by injecting the traffic into the router with a high rate causing the packets of legitimate traffic being dropped at the router queue due to congestion control. A reliable server should be able to block the attack traffic or limit the resource consumed by the possible illegitimate traffic, while guaranteeing the resources for the legitimate traffic. Therefore, how to protect the legitimate services from the illegitimate users during the DoS attacks is an important issue in Internet transmission.

An important component of secured network transmission is to employ intrusion detection to secure the service under network attacks. An intrusion detection system (IDS) can detect the possible attacks or abnormal network activities and signal an alert. Two major approaches are known as signature (misuse) detection and anomaly detection. In misuse detection [1][5][12], a learning algorithm is trained over the labeled training data. Misuse detection has the advantage of higher detection accuracy in detecting known attacks. The shortcoming is that it cannot detect previously unobserved attacks. On the other hand, anomaly detection approaches [7][11] build models from the normal data, and any deviation from the normal model in the new data is detected as anomaly. Anomaly detection has the advantage of detecting new types of attacks, while suffering from a high false alarms rates. To achieve a high detection performance, the false alarm rate should be kept as 5% or less.

A high performance Internet should have reliability, service availability, and resilience to the attacks. Most approaches addressing the service availability problem focus on how to defend against the DoS attacks [2][8][13][14]. In [2][8][14], the DDoS (distributed Denial-of-Service) attacks were viewed as a resource management problem and implemented via congestion control at the router, which involves network support. The approach proposed in [13] focuses on the availability of web services. It designed a practical DDoS defense system that can protect the availability of web services during severe attacks. Legitimate traffic are isolated and protected from the DDoS traffic when an attack occurs. Statistical approaches are proposed to defend against DDoS attacks [4][6]. [6] proposed statistical-based filtering to defend against and mitigate the effects of DDoS attacks. [4] proposed a DDoS defense architecture which supports distributed detection and automated on-line attack characterization. Our proposed framework integrates intrusion detection and resource management to provide reliable service under bandwidth attacks.

The paper is organized as follows. Section 2 gives the overview of the proposed framework. In Section 3, we present the simulation results and analysis. Conclusions are given in Section 4.

2 The Proposed Framework

This paper focuses on the integration of the attack detection and adaptive transmission management to mitigate the impact imposed on the network traffic by bandwidth attacks. To provide an efficient anomaly traffic detection scheme, several procedures in anomaly traffic detection are studied including the selections of anomaly detection method and features.

2.1 Anomaly Traffic Detection

2.1.1 Anomaly Detection Method

Anomaly detection assumes that the patterns not observed previously are treated as anomalies or outliers. A comparative study of anomaly detection schemes in network intrusion detection is conducted in [5]. As can be seen from [5], LOF has high detection rates when the false alarm rate is larger than 2%. For example, it can obtain a detection rate of 98.70% under a false alarm rate of 4% with a standard deviation of 0.42%. However, LOF has a very low detection rate when the false alarm rate is equal to or less than 2%. LOF is used as the anomaly detection approach in the proposed intrusion detection system. The idea of LOF is to assign a local outlier factor (LOF) to each object in the data set indicating its degree of being an outlier. Let k be the minimum number of objects that determine a density threshold. Using the Euclidean distance metric, the LOF of an object can be computed as follows.

1. For a specified value of the parameter k, compute the k-distance and find the k-distance neighborhood of each object P in the data set. The k-distance of P, denoted by k_distance(P), is the distance to its k^{th} nearest neighbor and the k-distance neighborhood of P is a set $N_{\mathrm{k}}(\mathrm{P})$ containing every object whose distance from P is not greater than the k-distance.

2. Compute the reachability distance for each object P with respect to an object O.

$$\mathrm{reach\_dist}_{\mathrm{k}}(\mathrm{P},\mathrm{O}) = \max\{d(\mathrm{P},\mathrm{O}), \mathrm{k\_distance}(\mathrm{O})\}. \qquad (1)$$

3. Compute the local reachability density of the object P.

$$\mathrm{lrd}_{\mathrm{k}}(\mathrm{P}) = \frac{|N_{\mathrm{k}}(\mathrm{P})|}{\sum\limits_{\mathrm{O}\in N_{\mathrm{k}}(\mathrm{P})} \mathrm{reach\_dist}_{\mathrm{k}}(\mathrm{P},\mathrm{O})}, \qquad (2)$$

where $|N_{\mathrm{k}}(\mathrm{P})|$ is the cardinality of $N_{\mathrm{k}}(\mathrm{P})$ indicating the number of objects in the k-distance neighborhood of P.

4. Compute LOF of an object P.

$$\mathrm{LOF}_{\mathrm{k}}(\mathrm{P}) = \frac{\sum\limits_{\mathrm{O}\in N_{\mathrm{k}}(\mathrm{P})} \mathrm{lrd}_{\mathrm{k}}(\mathrm{O})}{\mathrm{lrd}_{\mathrm{k}}(\mathrm{P})|N_{\mathrm{k}}(\mathrm{P})|}. \qquad (3)$$

An object P is considered an outlier if its LOF exceeds a threshold value.

2.1.2 Feature Extraction

Most of the anomaly detection algorithms need a set of normal data to train the model, and assume that the anomalies can be viewed as a pattern not observed before. The training data is composed of normal data randomly selected from all normal connections.

Training Data Set

For intrusion detection, KDD CUP 1999 data [3] is commonly used as the training data set. It is also used as the training data set in our simulation. KDD data set is designed for the purpose of building a network intrusion detector. The intrusion detector is actually a predictive model, which is capable of distinguishing between "bad" connections, called intrusions or attacks, and "good" normal connections. This data set contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment.

The training data set contains 494,021 connection records and the test data set contains 311,029 connection records. A connection is a sequence of packets with values of 41 features. Among the 41 features, there are 34 numerical features and 7 nominal features. Each connection is labeled as either normal or an attack. There are 22 attack types in the training data, which falls into 4 categories, DoS, probing, U2R and R2L. DoS and probing attacks involve many connections to some hosts in a very short period of time. R2L and U2R attacks normally involve only a single connection, and the attacks are embedded in the data portions of packets. The 41 features can be divided into basic features and derived features. There are two types of derived features. One is the content features within a connection suggested by domain knowledge. The other is the traffic features computed using a two-second time window. A complete list of the set of defined features for the connection records can be found in [3].

Feature Selection and Construction

Denial of Service (DoS) attacks usually make multiple connections to the same host within a short period of time. The connection can last from several seconds to several hours. The selection of features is an important issue in intrusion detection. The time-based and connection-based features are selected since they are more important for the detection of DoS and probing attacks. Most of the DoS and probing attacks may use hundreds of packets or connections, so time-based features can capture previous connections that have the same characteristics. Connection-based features are also necessary since there are some probing attacks that scan the hosts at a very slow speed, for example, one scan per minute or per hour.

The combination of important features can increase the detection rate while decreasing the computation cost, and thus improve the overall intrusion detection performance. In addition, different attack types have different important features [9]. For DoS attacks, some of the important features are $count$, srv_count, $dst\_host\_srv\_serror\_rate$, $serror\_rate$, $dst\_host\_same\_src\_port\_rate$, $source\ bytes$ and $destination\ bytes$. The selection of the features should be able to satisfy several requirements. First, it should achieve a high detection rate based on the training data set. Because of the limitations of the simulation tool (NS2) [10] used in our simulations, some of the features may not be able to be extracted efficiently. Hence, the selection of the features may have some limitations here. In a real network environment, the network traffic is collected via Tcpdump, and all the features should be able to be extracted.

In our simulations, 30 features are selected, among which are 7 nominal features and 23 numerical values. All these features are extracted from the connection information. 9 basic features are selected, such as $duration$, $protocol\_type$, $service$, src_bytes, dst_bytes, and $flag$. 10 content features are selected, such as num_failed_logins, $logged\_in$, $root\_shell$, $su\_attempted$, is_host_login, and is_guest_login. 11 traffic features computed using a two-second time window are selected, such as $count$, srv_count, $same\_srv\_rate$, $diff\_srv\_rate$, $srv\_diff\_host\_rate$, dst_host_count, $dst\_host\_srv\_count$, $dst\_host\_same\_src\_port\_rate$, $dst\_host\_same\_srv\_rate$, $dst\_host\_diff\_srv\_rate$, and $dst\_host\_srv\_diff\_host\_rate$.

2.2 Interaction with Resource Management

2.2.1 Protect against Possible Attacks

When abnormal traffic is detected by an IDS, some actions need to be taken automatically for the sake of security and resource management. Since connection-based intrusion detection is considered, when a connection is determined as abnormal, the connection will be disconnected and the traffic is discarded.

2.2.2 Procedures of Anomaly Traffic Detection

For every incoming connection to the server, a connection record is maintained for the purpose of monitoring the connection. Features are then extracted based on the basic or statistics information of the connections. The steps of the traffic monitoring and detection of attacks are listed below.

- Upon the request of a new connection, check if it is already in the attacker_node_list. If yes, return a Null socket and the connection from this node is rejected. If not, a connection is allowed.
- Upon the end of a current connection, statistics information is collected and features are extracted. Anomaly detection will be performed on the features to test if this is an anomaly or attack traffic.
- Upon the detection of anomaly traffic, the anomaly detection component will interact with the resource management component via signaling the detection of anomaly traffic. The resource management component will then take actions to protect the resources and services.
- Upon the detection of attack traffic, the original node of this connection is registered in the attacker_node_list. Current connections from this node will be disconnected. Hence, the traffic originating from this attacker node will be cut off.

3 Simulation and Analysis

3.1 Simulation Setup

NS2 [10] is used as the simulation tools. A simple scenario is used with an application server which provides FTP service and video streaming service. FTP file transmission is via TCP protocol. Video streaming is via UDP protocol. The network topology of the network is displayed in Figure 1. In this simulated network, an integrated environment with intrusion detection and resource management components at the server is built in NS2. The bottleneck link is the link between Router 3 and Router 4. Three FTP clients randomly connect to the application for file transmission. One of the 4 attack clients is selected randomly every time as the attacker tries to launch the DoS attack.

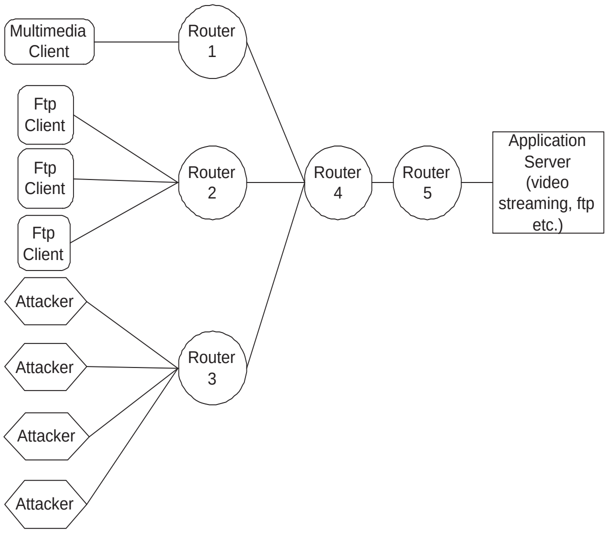

Figure 1. Network topology used in the simulation.

The purpose of the DoS attack generated by the attackers is to consume the bottleneck bandwidth. Since the DoS attack is launched via requesting FTP services, it will deplete the bandwidth from the server to the clients. Both of the legitimate traffic and attack traffic conform to TCP congestion control and share the bandwidth equally. In this way, the legitimate ftp clients and video streaming clients will compete for the bandwidth along the path from the server to the client with the attack ftp traffic. Both the attackers and the legitimate ftp users will request files from the server. Each attack can have multiple connections to the server. The file size for the legitimate clients is 1Mbps. The file size for the attacker ranges from 2,000 Bytes to 7,000 Bytes in the simulation. The larger file size specified for the attacks will result in a more severe attack load.

The detection rate varies with the false alarm rates. The detection rate is the ratio between the number of correctly detected attacks and the total number of attacks. On the other hand, the false alarm rate is the ratio between the number of normal connections that are incorrectly classified as attacks and the total number of normal connections. In the simulation, since there is no labeling for the network connections, it is unable to report directly the detection rate, false alarm rate and other evaluation metrics as reported in the DARPA'98 intrusion data set. Training data set consists of 4,000 records which are selected from the original KDD training data set.

3.2 Simulation Results

This simulation examines how the intrusion detection method deployed at the server side can effectively protect the bandwidth for the legitimate users. The performance metrics used in the simulations are

- Attack detection time;
- Connection number of attack traffic;
- Download time of attackers and legitimate users; and
- Throughput of the legitimate users.

The above performance metrics are studied under two cases: with intrusion detection and without intrusion detection. During this simulation, only the ftp service is running, so the impact of the attack traffic on the ftp flows can be studied. Performances are also compared under different attack loads. The file size used for the attack traffic ranges from 2,000 Bytes to 7,000 Bytes. The file size of 2,000 Bytes is used for the light attack; while the file size of 7,000 Bytes is for the severe attack. The first attack is launched at the 8^{th} second, the whole attack stops at the 35^{th} second, and the whole simulation ends at the 40^{th} second. For the legitimate clients, it starts a ftp connection every 5 seconds.

Table 1. First detection time information (in seconds) for the attacks.

File size (Bytes)	2,000	4,000	5,000	6,000	7,000
First detection time	4.50	4.80	4.82	5.65	5.89

The first detection time information (in seconds) for the attacks is shown in Table 1. This is the time when the first attack is detected by the host. It is reasonable to assume that the smaller the attack traffic load, the earlier the first detection of the attackers can be detected. This is because our intrusion detection component is performed on each connection after the connection ends. With lighter attack traffic, more connection information can be collected, which helps to detect the anomalous traffic.

With intrusion detection, the attack client will be registered in the server and no more connections from it will be granted, so the number of connections finished with intrusion detection should be smaller than that of without intrusion detection. The numbers of connections finished under

Table 2. Number of finished attack connections with and without intrusion detection.

File size (Bytes)	With Intrusion Detection	Without Intrusion Detection
2,000	283	881
4,000	288	881
5,000	292	881
6,000	299	875
7,000	301	867

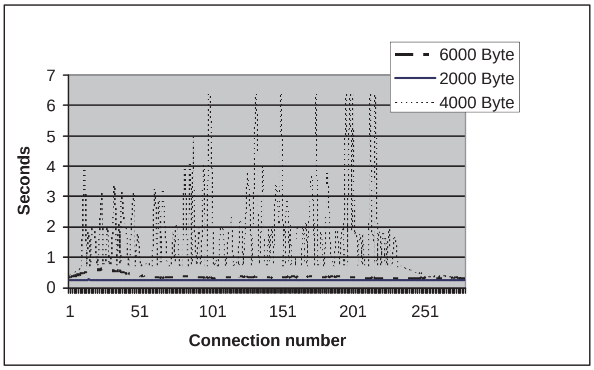

Figure 2. Download time for the connections of attack traffic with intrusion detection.

different attack loads with and without intrusion detection are shown in Table 2. It can be seen that the number of connections finished with intrusion detection is much smaller than that of without intrusion detection. Another observation is that the numbers of connections are quite comparable for the various file sizes with intrusion detection, and similarly they are quite comparable for the various file sizes without intrusion detection.

The download times for all the attack connections under different attack loads with intrusion detection are shown in Figure 2. Generally speaking, the download time increases with the file size. The download time for the file size of 4,000 Bytes is larger than that for the file size of 2,000 Bytes, but they are still close; while the download time for the file size of 6,000 Bytes is much larger than the others. Figure 3 gives the download time for attack connection without intrusion detection. The case of file size of 2,000 Bytes is ignored here since it is quite close to the case of file size of 4,000 Bytes. Hence, with the file size linearly increases, the download time for the traffic increases exponentially since the network is severely congested.

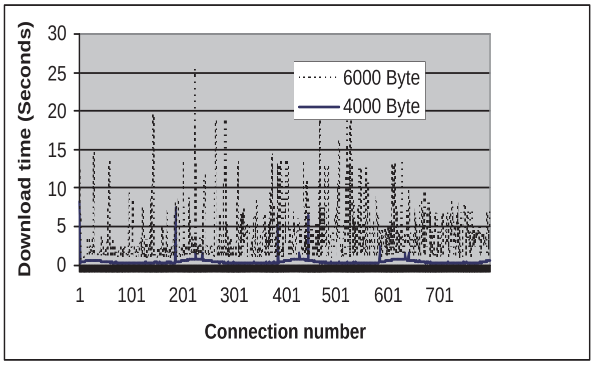

Figure 3. Download time for the connections of attack traffic without intrusion detection.

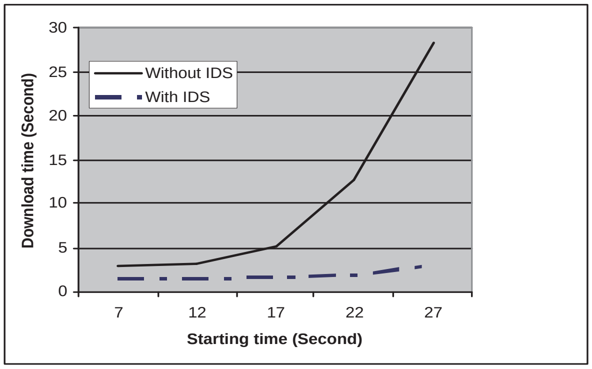

Figure 4. Download time for the connections of legitimate users.

Figure 4 gives the download time for the legitimate users with and without intrusion detection. It can be observed that the download times for all connections with intrusion detection are small and vary in a small range. On the other hand, the download times for the legitimate connections starting later increase exponentially because the network congestion is more severe. With the effect of bandwidth competition between the legitimate traffic and attack traffic, the bandwidth for the legitimate users is reduced, and at the same time, the download time increases.

The throughput of the legitimate files of each connection is also displayed in Figure 5. In both cases, the throughput for the connection decreases when the connection starts at a later time. When there is no intrusion detection, the throughput decreases drastically because of the severe network congestion. According to the simulations, even the linearly increased file size used by the attackers can make the network traffic increase drastically. With intrusion de-

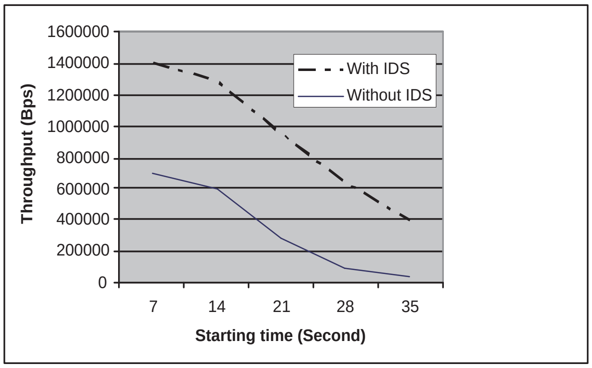

Figure 5. Throughput for connections of the legitimate users with and without intrusion detection.

tection, less attack traffic will be injected into the network to cause congestion, and hence bandwidth can be saved from the attack. Thus, the service time for the legitimate ftp file transfer can be greatly reduced.

4 Conclusion

Security has been an important issue in providing Internet services. Network attacks like DoS and probing consume lots of network resources, which is a threat to data transmission. Attack detection and active response to the attacks can mitigate the impact of DoS to provide better service for legitimate users. In this paper, we first investigate how to detect the attacks via a data mining based anomaly detection method. We then examine how to take an active response to the attack at the host so that the legitimate service can be guaranteed during the possible attacks. Intrusion detection, mitigation, and resource management are integrated. The prototype of the proposed system was implemented in NS2 to test the efficiency of the proposed framework. It includes attack traffic generation, features extraction, anomaly detection, and interaction between the attack detection and resource management components. Several performance metrics have been studied to show the efficiency of the proposed framework in the detection of anomaly traffic. Via the early detection of incipient DoS and the interaction with the resource management component, legitimate services can be better protected.

References

[1] V. Paxson. Bro: A system for detecting network intruders in real-time. In *Proceedings of 7th USENIX Security Symposium*, 1998.

[2] S. Chen and Q. Song. Perimeter-based defense against high bandwidth DDoS attacks. *IEEE Transactions on Parallel and Distributed Systems*, 16(6):526–537, July 2005.

[3] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (available on 2007).

[4] Y. Kim, W.C. Lau, M.C. Kau, M.C. Chuah, and J.H. Chao. Packetscore: Statistics-based overload control against distributed denial-of-service attacks. In *Proceedings of IEEE Infocom 2004*, pages 2594–2604, 2004.

[5] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *SIAM 2003*, 2003.

[6] Q. Li, E.-C. Chang, and M.C. Chan. On the effectiveness of DDoS attacks on statistical filtering. In *IEEE Infocom 2005*, volume 2, pages 1373–1383, March 2005.

[7] M. Mahoney and P. Chan. Learning rules for anomaly detection of hostile network traffic. In *Proceedings of the Third IEEE International Conference on Data Mining*, pages 601–604, 2003.

[8] R. Mahajan, S. Bellovin, S. Floyd, J. Joannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregate in the network. *Computer Communications Review*, 32(3):62–73, July 2002.

[9] S. Mukkamala, A. H. Sung, and A. Abraham. *Enhancing Computer Security with Smart Technology*, chapter Cyber Security Challenges: Designing Efficient Intrusion Detection Systems and Antivirus Tools, pages 125–161. CRC Press, USA, 2005.

[10] http://www.isi.edu/nsnam/ns/. Network simulator, 2007, (available on 2007).

[11] M.-L. Shyu, K. Sarinnapakorn, I. Kuruppu-Appuhamilage, S.-C. Chen, L. Chang, and T. Goldring. Handling nominal features in anomaly intrusion detection problems. In *15th International Workshop on Research Issues on Data Engineering (RIDE Workshop of Stream Data Mining and Applications RIDE-SDMA '2005), in conjunction with The 21st International Conference on Data Engineering (ICDE 2005)*, pages 55–62, Tokyo, Japan, April 3-4 2005.

[12] Snort. The open source network intrusion detection system. http://www.snort.org, (available on 2007).

[13] J. Xun and W. Lee. Sustaining availability of web services under distributed denial of service attacks. *IEEE Transactions on Computers*, 52(2):195–208, February 2003.

[14] D. K.Y. Yau, J.C.S. Lui, F. Liang, and Y. Yam. Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. *IEEE/ACM Transactions on Networking*, 13(1):29–42, February 2005.