Detecting DoS and DDoS Attacks using Chi-Square

Fang-Yie Leu and Chia-Chi Pai
Department of Computer Science, Tunghai University, Taiwan
leufy@thu.edu.tw
g942922@thu.edu.tw

Abstract

In this paper, we propose an agent-based distributed intrusion detection architecture, which detects DoS/DDoS attacks by comparing source IP addresses’ normal and current connection frequencies. First, we collect source IPs’ packet statistics to obtain their normal packet distribution. When current statistics suddenly increase, very often it is an attack. Experimental results show that this approach can effectively detect DoS/DDoS attacks.

1. Introduction

Nowadays, users can easily access and download network attack tools from the Internet, which often provide friendly interfaces. Therefore, even a naïve user can also launch a large scale DoS or DDoS attack. DoS is a type of attack in which a hacker issues a huge amount of packets to congeal specific servers’ services, consequently blocking legitimate users from normal access to the services. DDoS attacks are another form of DoS in which a host or hosts suffer from receiving a huge amount of packets issued by Zombies [1].

In this paper, we propose an agent-based distributed security system, called Agent-based Instruction Detection System (AIDS), to detect DoS and DDoS attacks. AIDS uses mobile agents to decentralize tasks of data analysis, and employs distributed components to reduce workload of detection tasks.

2. Related Work

The Distributed Intrusion Detection System architecture [2] combines distributed monitoring and data reduction with centralized data analysis. But it did not scale well for large networks. The Autonomous Agents for Intrusion Detection (AAFID) [3] made use of multiple layers of agents organized in a hierarchical structure with each layer performing a set of intrusion detection tasks. However, AAFID uses only static agents and is deprived of some of the benefits mobile agents can offer.

3. System Architecture

AIDS system architecture as shown in Figure 1 consists of four main components, including event monitoring subsystems, backup subsystems, a duty center and a black list database.

3.1. Event Monitoring Subsystems

An event monitoring subsystem as shown in Figure 1 comprises event analyzers and an event collector, which are employed to protect geographically concentrated subnets. A building with several subnets owned by different departments or the same department is an example.

An event analyzer collects source and destination addresses for packets sent to hosts that it protects through a switch having a mirror port, and counts number of packets each sender (source IP address) sends to a specific host or subnet. A packet \( P \) flowing through the switch will be duplicated. The duplicated packet is then sent out via the mirror port from which an event analyzer gathers all its detecting packets. \( P \) will continue its journey to destination. In addition, an event analyzer creates a source-IP distribution table, which as shown in...
Table 1 has four main attributes, source-IP, port #, Group # and packet-information (packet-inf for short), where packet-inf is used to count number of and accumulates size of packets that P’s sender has sent during a specific period of time, and with which we can detect DoS/DDoS attacks, and identify who is issuing the attacks. It periodically, once per ten second, dispatches a mobile agent to deliver this table to its event collector to accumulate packets sent to the subnets in underlying building.

<table>
<thead>
<tr>
<th>Source-IP</th>
<th>Port #</th>
<th>Group #</th>
<th>past 7-day-count</th>
<th>past 7-day-size</th>
<th>one-week sum-up-size</th>
<th>current-day-count</th>
<th>current-day-size</th>
<th>Past 10-sec-count</th>
<th>Past 10-sec-size</th>
</tr>
</thead>
</table>

Table 1 A source-IP distribution table

Before detection, we first collect ordinary packets for subunits of a geographically concentrated unit for one week to establish a baseline profile which is updated daily. After each update, we calculate chi-square value \( \chi^2 \) for each group \( i \), \( i=0,1,2,…,12 \), as their new baseline. \( x_{kj} \) may be either \( PD^2 \) and \( xD^2 \) for group \( j \), when \( k \) is the th group no., we accept the null hypothesis, (or \( 1.026 \) away from \( \chi^2 \)).

Algorithm 1: Establishing a source-IP distribution table

Input: An incoming packet \( P \) with source IP, S-IP, and destination IP, D-IP
Output: update S-IP’s information in T_D-IP

If (S-IP is already in T_D-IP, e.g., tuple t)
| Packet-inf |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| one-week sum-up-size | current-day-count | current-day-size | Past 10-sec-count | Past 10-sec-size |

Algorithm 2: Establishing a baseline profile-for count

Input: a source-IP distribution table \( T \)
Output: tuples in \( T \) are classified into 13 groups

If (timer times out) /* reset past-10-sec-count*/

For (i=1; i<10; i++)

If (i=10, i<10; i++)

Shift \( t.i \)’s day to \( (i+1) \)th day-count;
Shift current-day-count to \( t.i \)’s day-count;
For (each tuple \( t \) in \( T \))

\( t \).one-week sum-up-count = \( \sum_{i=1}^{7} t.i \)’s day-count;

Sort tuples in \( T \) on “one-week sum-up-count” field;
The IP ranked top one is group 0;
The IPs ranked top 2nd and 3rd are group 1;
The IPs ranked top 4th to 7th are group 2;

The IPs ranked top 7(1)th to (2)2-th are group 11;
The IPs ranked top 212 and up are group 12;
For (each tuple \( q \) in \( T \))

Fill in “q.group-No” field the group number to which \( q \) belongs
Set timer=24 hr;

In algorithm 2, a baseline profile is established for count. The baseline profile for size is also established by the similar algorithm with count being replaced by size. To detect DoS/DDos attacks, let

\( N_i \text{count} \) (\( N_i \text{size} \)) is number of packets (accumulated packet size) that group \( i \) has currently received in past 10 seconds when there is no attacks. Let

\( x_{i,\text{count}}^2 = \frac{(\sum x_{i,\text{count}} - G_{i,\text{count}})^2}{G_{i,\text{count}}} \) and \( x_{i,\text{size}}^2 = \frac{\sum (x_{i,\text{size}} - G_{i,\text{size}})^2}{G_{i,\text{size}}} \)

where \( N_i \text{count} \) (\( N_i \text{size} \)) is number of packets (accumulated packet size) that group \( i \) has currently received in past 10 seconds, \( n=13 \), and \( df=(n-1=12) \) is degree of freedom. In this research, we choose significant standard \( \alpha = 0.05 \), so \( x_{0.05}^2 = 21.026 \).

If \( x_i^2 \leq x_{12,0.05}^2 \), we accept the null hypothesis \( H_0 \), indicating that there is no attack where \( x_i^2 \) may be either \( x_i^2 \text{count} \) or \( x_i^2 \text{size} \). However, if \( x_i^2 \text{count} > x_{12,0.05}^2 \) (or \( x_i^2 \text{size} > x_{12,0.05}^2 \) ), we reject \( H_0 \), which means its alternative hypothesis \( H_1 \) is true, showing that there is a suspected resource consumption (bandwidth consumption) attack. The algorithm invoked to detect resource consumption DoS/DDoS attacks is shown in algorithm 3.

Algorithm 3: detecting resource consumption DoS/DDoS attacks with chi-square method by an event analyzer

Input: a source-IP distribution table \( T \); a baseline profile
Output: whether or not a subnet or subnets protected by an event analyzer are under a DoS/DDoS attack

1. Att=false;
2. If (timer times out) /* initial value is 10 sec*/
   If \( (x^2 - \text{count} \geq (x^2_{12,\text{count}} + \text{threshold}) ) \)
   \{For (i=0; i<=12; i++)
   \text{If} \left( \frac{\sum_{i=1}^{m_i} 10-\text{sec-count}}{\sum_{i=1}^{m_i} 10-\text{sec-count}} \geq 80\% \right) \}
   \text{Choose k IPs whose past 10-sec-counts are the highest as the hackers, where}
   \text{m}_i \text{ is number of source IPs in group i;}
   \text{Send an alert message to duty center and administrator to show that there is a resource}
   \text{consumption DoS/DDoS attack;}
   \text{Send source IP addresses that issue the attacks to the black list database.}\}

The algorithm for detecting bandwidth consumption attacks is similar to algorithm 3 with count being replaced by size.

An event collector on receiving source-IP distribution tables from all its event analyzers once per 10 seconds accumulates the table contents to its source-IP accumulation table, of which the table structure is similar to that of a source-IP distribution table, and checks to see whether the chi-square values of current network traffic (including size and count) for its 13 groups significantly exceed event-collector-level thresholds or not. If yes, there is an attack which is attacking subnets protected by some of its event analyzers with low-density traffic. And regardless of yes or no, it dispatches a mobile agent to send its source-IP accumulation table to the duty center.

3.2. Duty Center

The duty center, as the coordinator of AIDS, further detects whether or not there is a DoS/DDoS attack which is attacking the protected system. The duty center builds a source-IP integration table, of which the table structure is similar to that of a source-IP distribution table, to detect attacks. If yes, the duty center dispatches a mobile agent to record attackers’ information also in the black list database. With the database, a firewall can accordingly filter out packets issued by known hackers. Algorithm 4 shows the detection details of the duty center.

Algorithm 4: detection process of the duty center
Input: source-IP accumulation tables received periodically
Output: whether a network management unit, e.g., a university/company, is under a DoS/DDoS attack
\{If (timer of an event collector E times out)
\{Dispatches a mobile agent as a backup agent to check status of E;
If (E fails)
\{The agent chooses the host with the highest performance, e.g., \( h_j \), from the corresponding backup system, and asks the host acting as an event collector to take over for E;
Change \( h_j \)'s network interface (i.e., network card) into promiscuous mode to filter packets originally sent to E;
Else asks E to send content of its source-IP accumulation table to the duty center;
2. Integrate all source-IP accumulation tables received to generate its source-IP integration table;
3. Detect whether there is a DoS/DDoS attack by calculating \( G_{i\text{-count}}, G_{i\text{-size}}, x^2_{\text{count}} \) and \( x^2_{\text{size}} \) at duty-center level;
4. If (yes)
\{Send a message to administrators;
Dispatch an update agent to deliver hacking information to the black list database.\}\}

3.3. Backup Subsystem

When an event analyzer is under a DoS/DDoS attack and loses its detection capability, its event collector dispatches a backup agent to the corresponding backup subsystem to choose a host and requests the host acting as an event analyzer to substitute for the attacked one. The process of selecting a backup host by an event collector is as follows.

Algorithm 5: choosing a backup host for an attacked event analyzer
Input: a set of m hosts \( H = \{ h_1, h_2, h_3, \ldots, h_m \} \) in the underlying geographically concentrated unit.
Output: the chosen host acts as an event analyzer
\{Change \( h_i \)'s network interface (i.e., network card) into promiscuous mode to filter packets sent to the protected subnet or subnets;\}

The algorithm that the duty center invokes to select a backup host as an event collector is described in algorithm 4.

4. Experiments and Discussion

Our experimental environment comprises computer rooms of Science-Technology, Science-college and Engineering-college Buildings in Tunghai University. We used 40 clients as attackers and one node as victim. Wireshark (version 0.99.5) was installed on the victim to gather traffic issued by the 40 attackers, and Trinoo was employed as the attack tool which uses 27444 port to send attack packets.
In experiment 1, we issued 323,201 packets/sec by several computers in some groups, e.g., group 0 sends 18.75% of attack packets, group 1 delivers 34.38% of attack packets, etc. In experiment 2, 700,002 packets/sec were issued also by several groups.

Table 2 Packet statistics collected in experiment 1 for an event analyzer’s source-IP distribution table, and the chi-square statistics

<table>
<thead>
<tr>
<th>Group</th>
<th>Source-IP</th>
<th>past 10 sec count</th>
<th>one-week average</th>
<th>current-day count</th>
<th>Gs</th>
<th>Non-normal packets</th>
<th>Percent</th>
<th>Chi-square</th>
<th>Degree for %</th>
<th>Chi-square for amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>140.128.201.1</td>
<td>3572</td>
<td>274,038</td>
<td>204,456</td>
<td>1.07</td>
<td>204,456</td>
<td>0.010</td>
<td>1.36</td>
<td>0.05</td>
<td>1.36</td>
</tr>
<tr>
<td>2</td>
<td>140.128.201.1</td>
<td>3572</td>
<td>274,038</td>
<td>204,456</td>
<td>1.07</td>
<td>204,456</td>
<td>0.010</td>
<td>1.36</td>
<td>0.05</td>
<td>1.36</td>
</tr>
<tr>
<td>3</td>
<td>140.128.201.1</td>
<td>3572</td>
<td>274,038</td>
<td>204,456</td>
<td>1.07</td>
<td>204,456</td>
<td>0.010</td>
<td>1.36</td>
<td>0.05</td>
<td>1.36</td>
</tr>
</tbody>
</table>

In experiment 2, the packet statistics listed in the source-IP accumulation table (Table 4) were gathered by accumulating several source-IP distribution tables. Its “chi-square for percentage” is 86.58 (>21.026). So, we can also conclude that there is a DoS/DDoS attack.

For accuracy test, we gather 100 times of normal and 100 times of attack traffic of 10 seconds. The attack densities range from 500 to 15,000 packets/sec. Table 5 shows the results. The errors (false-positives) result from the fact that at some time G% and N% had similar percentages, and were treated as normal traffic.

Table 5 Detection Accuracy of DoS/DDoS attacks

Table 3 Packet statistics collected in experiment 1 for an event-collector’s source-IP accumulation table and the chi-square statistics

<table>
<thead>
<tr>
<th>Source-IP</th>
<th>past 10 sec count</th>
<th>one-week average</th>
<th>current-day count</th>
<th>Gs</th>
<th>Percent</th>
<th>Chi-square</th>
</tr>
</thead>
<tbody>
<tr>
<td>140.128.201.1</td>
<td>3572</td>
<td>274,038</td>
<td>204,456</td>
<td>1.07</td>
<td>204,456</td>
<td>0.010</td>
</tr>
<tr>
<td>140.128.201.1</td>
<td>3572</td>
<td>274,038</td>
<td>204,456</td>
<td>1.07</td>
<td>204,456</td>
<td>0.010</td>
</tr>
<tr>
<td>140.128.201.1</td>
<td>3572</td>
<td>274,038</td>
<td>204,456</td>
<td>1.07</td>
<td>204,456</td>
<td>0.010</td>
</tr>
</tbody>
</table>

In the future, we would like to study AIDS’s system behavior and reliability, and develop the behavior and reliability models. So, users can accordingly predict AIDS’s system behavior and reliability before using it.

6. References

