DDoS detection and traceback with decision tree and grey relational analysis

Yi-Chi Wu

Department of Computer Science,
National Chiao Tung University,
Hsinchu 30010, Taiwan
Fax: +886-3-5721490
E-mail: yoshieason@gmail.com

Huei-Ru Tseng

Industrial Technology Research Institute,
Hsinchu, 31040, Taiwan
E-mail: hueiru@itri.org.tw

Wuu Yang* and Rong-Hong Jan

Department of Computer Science,
National Chiao Tung University,
Hsinchu 30010, Taiwan
Fax: +886-3-5721490
E-mail: wuuyang@cs.nctu.edu.tw
E-mail: rhjan@cs.nctu.edu.tw
*Corresponding author

Abstract: In Distributed Denial-of-Service (DDoS) Attack, an attacker breaks into many innocent computers (called zombies). Then, the attacker sends a large number of packets from zombies to a server, to prevent the server from conducting normal business operations. We design a DDoS-detection system based on a decision-tree technique and, after detecting an attack, to trace back to the attacker's locations with a traffic-flow pattern-matching technique. Our system could detect DDoS attacks with the false positive ratio about 1.2–2.4%, false negative ratio about 2–10%, and find the attack paths in traceback with the false negative rate 8–12% and false positive rate 12–14%.

Keywords: DDoS detection; attacker traceback; decision tree; grey relational analysis.

Reference to this paper should be made as follows: Wu, Y-C., Tseng, H-R., Yang, W. and Jan, R.H. (2011) 'DDoS detection and traceback with decision tree and grey relational analysis', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 7, No. 2, pp.121–136.

Biographical notes: Yi-Chi Wu received the MS in Institute of Computer Science and Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2008. His research interests include artificial intelligence and network security.

Huei-Ru Tseng received the BS and MS in Information Management from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2002 and 2004, respectively. She obtained her PhD in Computer Science from National Chiao Tung University, Hsinchu, Taiwan, in 2010. Since then, she has worked in Industrial Techonology Research Instititute. Her research interests include vehicular ad hoc networks, wireless networks, network security and cryptography.

Wuu Yang received the BS in Computer Science from National Taiwan University in 1982 and the MS and PhD in Computer Science from the University of Wisconsin at Madison in 1987 and 1990, respectively. Currently, he is a Professor in the National Chiao Tung University, Taiwan. His current research interests include Java and network security, programming languages and compilers and attribute grammars. He is also very interested in the study of human languages and human intelligence.

Rong-Hong Jan received the BS and MS in Industrial Engineering, and the PhD in Computer Science from National Tsing Hua University, Taiwan, in 1979, 1983 and 1987, respectively. He was a Visiting Fellow in the Sloan School of Management, MIT, MA, from January 1986 to 1987. He joined the Department of Computer Science, National Chiao Tung University, in 1987, where he is currently a Professor and Associate Dean. During 1991–1992, he was a Visiting Associate Professor in the Department of Computer Science, University of Maryland, College Park, MD. His research interests include wireless networks, mobile computing, distributed systems, network security and network reliability.

1 Introduction

With the proliferation of computer networks come many kinds of network attacks. Among them, the DDoS attacks (Zaroo, 2002; Mirkovic and Reiher, 2004; Douligeris and Mitrokotsa, 2004; Peng et al., 2007) have caused serious economic loss. Therefore, effective and efficient protection systems are urgently needed. Denial-of-service attacks, as the term suggests, attempt to deny legitimate users the services that the servers provide.

Because an attacker could modify the source IP addresses in the packets (i.e., IP spoofing), tracing back the origin of an attack becomes very difficult. We design a system that detects DDoS attacks quickly and traces back the origins of DDoS attacks quite accurately. The characteristics of our system include:

- there is no need to modify existing protocols (e.g., TCP/IP)
- the set-up procedures on routers are simple
- the system can accommodate novel attacks in the future
- the system can fit any network topology
- the traceback procedure is efficient.

In this paper, we focus on the flooding-based attack aiming at layer 3/layer 4 in the OSI 7-layer model. Our system basically consists of two subsystems, the *protection agent* located only in victim and the *sentinels* located in routers. Both protection agent and sentinels collect all the packets passing them and retrieve the information in network layer 3/layer 4 from those packets, then aggregating those retrieved information for the purpose of detection and further traceback.

The main concept of our proposed DDoS detection is based on deciding the traffic flow pattern under the situation without attack and the one under different attacks. Therefore, the detection of attack from normal situation could be viewed as the classification problem and we propose 15 different attributes, which not only monitor the incoming/outgoing packet/bytes rate but also compile the TCP SYN and ACK flag rate, to describe the traffic flow pattern. We apply the decision tree (C4.5) technique taking these attributes as the tests to detect abnormal traffic flow (Peng et al., 2007). Then, we use a novel traffic pattern, matching procedure to identify the traffic flow that is similar to the attack flow and, based on this similarity, to trace back the origin of an attack.

The rest of this paper is organised as follows. In Section 2, we discuss the existing detection and traceback mechanisms. Next, we introduce the architecture of our system in Section 3. In Section 4, our proposed detection and traceback method is presented. In Section 5, the experiment results indicate that our proposed system is capable of detecting the attacks and tracing them back with high accuracy. Finally, we will conclude our paper in Section 6.

2 Related work

In this section, we will introduce several DDoS detection and traceback mechanisms (Noh et al., 2003; Liu and Uddin, 2005; Mirkovic et al., 2003; Mirkovic and Reiher, 2005; Savage et al., 2000; Stone, 2000; Burch, 2000; Bellovin et al., 2001) nowadays and briefly review how they work. We also summarise the characterisation of each mechanism in Table 1.

2.1 Related DDoS detection mechanisms

The existing DDoS detection mechanisms can be divided into two categories depending on the locations of the DDoS attack detection systems. One is victim based, in which the detection system is deployed close to the victim. The other is source based, in which the detection system is placed close to the attack source.

2.1.1 Victim-based detection

The system applied victim-based detection mechanism is located near the victim. The advantage of this method is easy and quick to detect the attack. But the mechanism is unable to mitigate the congestion on the attack path.

- *Machine learning*: Machine-learning techniques have been widely applied for DDoS attack detection. Traffic Rate Analysis (TRA) monitors the distribution rate of flags in TCP packets (Noh et al., 2003). This method calculates two metrics, TCP flag rates and protocol rates, as the criteria to

trigger alarms. Then, it applies machine-learning methods to identify the traffic patterns under attack. Another system monitors not only the network traffic but also the utilisation of resources (Noh et al., 2003). DDoS detection in this system is based on back-propagation Artificial Neural Network (ANN) and Bayesian classifier.

- *Statistical models*: In Liu and Uddin (2005), the non-parametric cumulative sum (CUSUM) algorithm based on statistical models is used for detection. It made use of a model for SYN and SYN-ACK packets. Whenever the number of collected SYN packets is larger than SYN-ACK packets, a positive value of CUSUM is derived. A large CUSUM indicates a large number of SYN packets without the pairing SYN-ACK packets. This usually implies a DDoS attack.

2.1.2 Source-based detection

Source-based detection attempts to deploy the detection systems as close to the attacker as possible. Therefore, the placement of the detection system would be on

Table 1 The comparison of DDoS detection and traceback mechanisms

Scheme	*Type*	*Advantages*	*Disadvantages*
DDoS detection mechanisms			
Noh et al. (2003)	Victim-based	• Apply machine-learning methods to identify the traffic patterns under attack	• Apply only on the rate of appearance of specific flags in the packets' headers (Öke and Loukas, 2007)
Liu and Uddin (2005)	Victim-based	• Statistical-based SYN-flooding detection • Achieve high detection accuracy while maintaining low computation overhead	• Difficult to anticipate a priori the best values for threshold parameters (Berral et al., 2008)
Mirkovic and Reiher (2005)	Source-based	• Detect the attacks and limit the traffic flowing into the intranet • Rely on edge routers to identify the sources of the flood of attacking packet	• Suffer from the scalability problem and difficulty of attack traffic identification (Fallah, 2010)
DDoS traceback mechanisms			
Stone (2000)	Link-testing (Input debugging)	• The victim reports the attack signature to the network operator	• Heavy management overhead (Savage et al., 2000)
		• Legitimate traffic would not be affected	• Heavy router overhead (Savage et al., 2000)
Burch (2000)	Link-testing (controlled flooding)	• The victim works with the closest routers to attack every link in the routers	• Legitimate traffic would be affected
		• Does not require the collection of attack signatures	• Heavy network overhead (Savage et al., 2000)
Savage et al. (2000)	Packet marking	• Record IP address or ID on the unused or rarely used fields in the IP packets	• High computation overhead for the victim to reconstruct the attack paths (Qu et al., 2005)
		• The management, network, and router overhead is quite low	• Give a large number of false positives when the attack originates from multiple attackers (Qu et al., 2005)
Bellovin et al. (2001)	ICMP traceback	• Special ICMP packet transmission • Reconstruct the attack path with ease	• The traceback could fail since the ICMP packets are sometimes filtered out • The input debugging capability that ICMP traceback message relies on may be not available in some router architectures (Savage et al., 2000)

the edge routers. In Mirkovic et al. (2003), Mirkovic et al. introduce five observations to build an effective source-based detection system. D-WARD (Mirkovic and Reiher, 2005), a famous source-based detection system, is deployed at the gateway router to detect the attacks and to limit the traffic flowing into the intranet. D-WARD consists of two components, which are the observation and throttling components. The observation component detects the abnormal traffic by traffic statistics. When an abnormal traffic is detected, the throttling component then adjusts the traffic rate in source routers to limit the attack traffic.

2.2 Related traceback mechanisms

Traceback is also an important defence against DDoS attacks. Because of IP spoofing, the source IP address in a packet is of little use in traceback. There are several IP traceback methods introduced in Savage et al. (2000).

2.2.1 Link testing

Link testing starts from the router closest to the victim and tests the upstream links of the router to identify the incoming link of the attack packets. Then, the same procedure is repeated to identify the upstream routers on the attack path one by one, until the origin of the attack is located.

There are two variations of link testing: input debugging (Stone, 2000) and controlled flooding (Burch, 2000). Input debugging requires that the victim reports the attack signature to the network operator. This results in heavy management overhead when the traceback runs across AS-level networks. Controlled flooding is a traceback method that applies the DoS technique. The victim works with the closest routers to attack every link in the routers. If the rate in receiving malicious packets in a particular link drops all of a sudden, then that link is possibly on the attack path. Although this method does not require the collection of attack signatures, this method itself is an attack towards the routers. Therefore, legitimate traffic would be affected as well.

2.2.2 Packet marking

In packet marking, (a part of) routers' IP addresses or ids are recorded on the unused or rarely used fields in the IP packets. There are many variations of packet marking. PPM node-append, PPM node-sampling and PPM edge-sampling are the most popular packet marking methods (Savage et al., 2000).

2.2.3 ICMP traceback

ICMP traceback (Bellovin et al., 2001) is a traceback method that makes use of ICMP packets. ICMP traceback is similar to packet marking. A designated router, called the iTrace, probabilistically copies a part of the contents of the received packets into a special ICMP packet, which also contains the addresses of the previous and the following routers. iTrace sends this special ICMP packet to the source and destination of the original packet. The victim could easily reconstruct the attack path according to the special ICMP packets. However, ICMP traceback could fail since the ICMP packets are sometimes filtered out. (Some routers simply throw away all ICMP packets to prevent ICMP flooding attacks.)

3 Proposed system

In this section, we will present the proposed detection and traceback system. It includes an artificial intelligence-based (AI-based) classifier for DDoS detection and a traffic-flow pattern matcher (Kim and Helmy, 2005) for comparing traffic signatures and for tracing back DDoS attacks.

3.1 System architecture

Our system consists of two components: *protection agents* and *sentinels*. A protection agent is deployed at the victim site for the detection purpose and the sentinels are deployed at all the routers for the traceback purpose. The overall organisation is shown in Figure 1. The links between the protection agent and the sentinels are secured tunnels, which make use of port forwarding in SSH-2 (secure shell protocol version 2) for preventing man-in-the-middle attacks.

3.2 System modules

In this subsection, we will introduce the components within the protection agent and sentinels.

3.2.1 Protection agent

The protection agent is the control centre of the entire system. The DDoS attack detection and attack path reconstruction are all handled in the protection agent. A protection agent consists of four components: a packet aggregator (to aggregate the traffic signatures), a message manager (to construct the SSH-tunnel and handle communication between the protection agent and sentinels), a DDoS attack detection module and a traceback module. The DDoS attack detection module includes the decision tree and rules. The message manager resides in the traceback module; the traceback module handles attack path reconstruction. Figure 2 presents the overview of the protection agent. The procedure and the flow chart of the protection agent is shown here and depicted in Figure 3.

1 Obtain the signature of the current traffic flow.
2 The detection module determines if an attack is going on based on the current traffic signature.
3 If there is no attack, the agent stores the traffic signature into the repository.

Figure 1 Overall organisation of our system (see online version for colours)

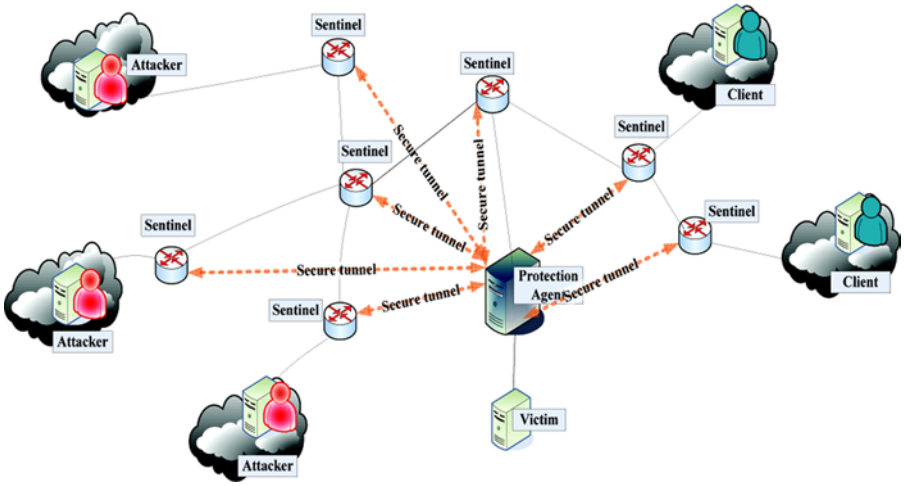


Figure 2 Modules in the protection agent (see online version for colours)

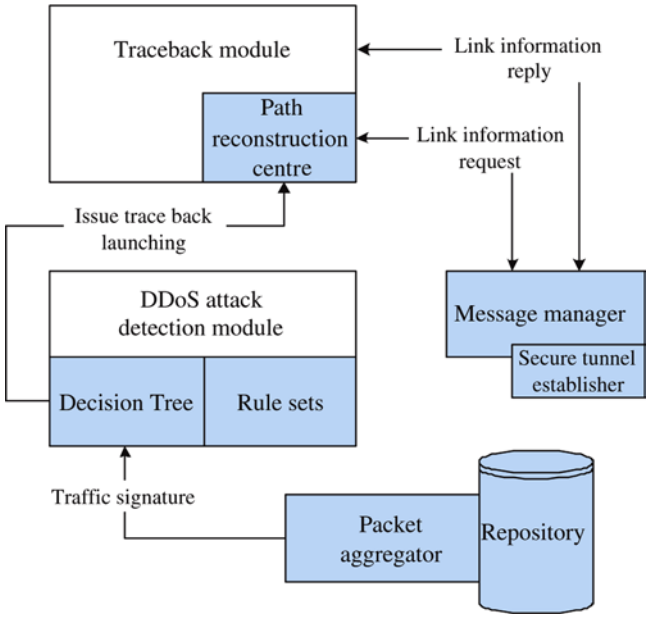
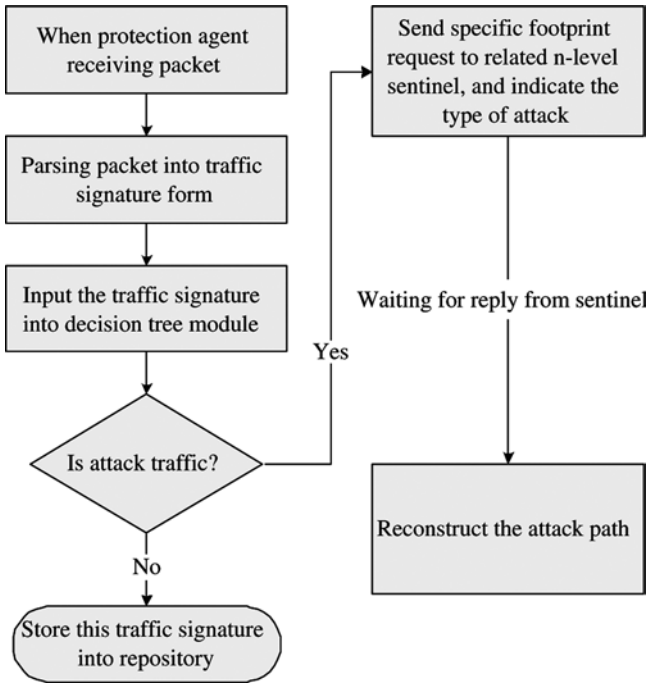


Figure 3 Operating flow chart in the protection agent



4 If there is an attack, then

   4.1 The agent issues a traceback command to the upstream sentinel.

   4.2 Wait until enough connection information (which contains the IP addresses of the two ends of the link and the distances from the victim) is collected.

   4.3 Construct the attack path with the collected connection information.

5 Go to step 1.

3.2.2 Packet aggregator

The packet aggregator computes a *traffic signature* based on all the packets passing through. The traffic signature is used for detection and traceback. With the help of pcap (http://www.tcpdump.org, 2004), our system captures all incoming and outgoing packets. For each packet, the packet header from layer 3 to layer 4 is extracted for cross-layer monitoring. The header information is used to compute a traffic signature, whose format is shown in Table 2. Our system generates one traffic signature per minute.

The traffic signatures are stored in the traffic signature repository with timestamps of the packet arriving time. A Bloom filter (Bloom, 1970) is used to reduce the memory overhead while collecting the IP addresses. The Bloom filter computes k (which is the number of hash functions used in the bloom filter) distinct digests for each IP address with independent hash functions, and uses the n-bit results to index into a 2^n-bit array. An example of a Bloom filter is depicted in Figure 4. We implemented two basic hash functions, SAX and SDBM, as the default

Table 2 Format of a traffic signature

	Attributes	*Value*
1	Incoming packet count per t minute(s)	Numeric
2	Incoming octets per t minute(s)	Numeric
3	No. of incoming TCP packets per t minute(s)	Numeric
4	No. of incoming UDP packets per t minute(s)	Numeric
5	No. of incoming ICMP packets per t minute(s)	Numeric
6	No. of incoming unknown-protocol packets per t minute(s)	Numeric
7	No. of incoming IP addresses/No. of outgoing IP addresses	Numeric
8	Outgoing packet count per t minute(s)	Numeric
9	Outgoing octets per t minute(s)	Numeric
10	No. of outgoing TCP packets per t minute(s)	Numeric
11	No. of outgoing UDP packets per t minute(s)	Numeric
12	No. of outgoing ICMP packets per t minute(s)	Numeric
13	No. of outgoing unknown-protocol packets per t minute(s)	Numeric
14	No. of incoming TCP SYN packets/No. of incoming TCP ACK packets	Numeric
15	No. of incoming IP addresses/No. of incoming packets per t minute(s)	Numeric
16	Time interval	Bed-time, morning, afternoon, night

Figure 4 Bloom filter (see online version for colours)

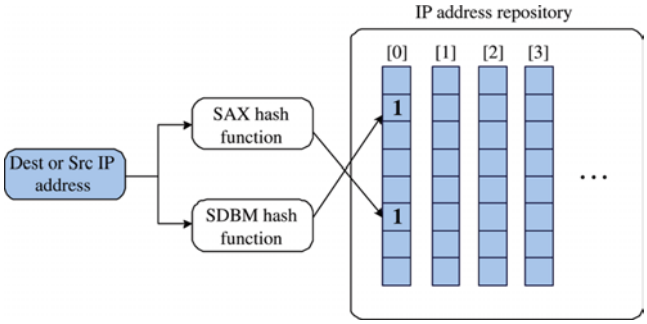
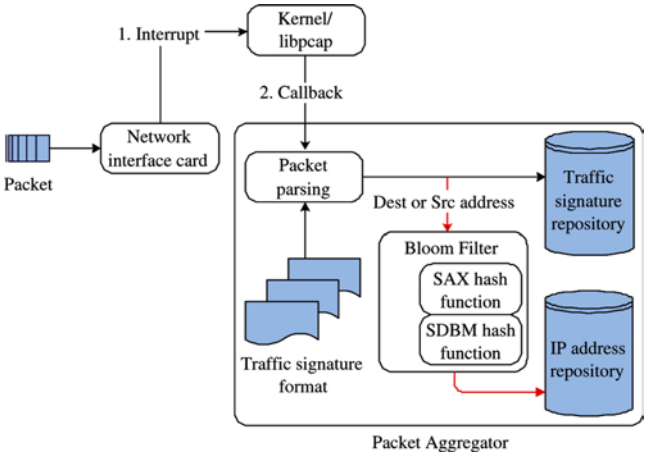


Figure 5 Packet aggregator (see online version for colours)



hash functions in the Bloom filter. The flow chart for packet processing is depicted in Figure 5. With the help of the Bloom filter, we could reduce the 32-bit IP address into 2 bits.

3.2.3 *n-hop sentinels*

The n-hop sentinel is located in a router. The 'n' represents the number of hops from the victim. There are four components within the sentinel: message manager, simplified packet aggregator, traffic-pattern matching module and traffic signature repository. Figure 6 presents the overview of the sentinel. A sentinel aggregates the headers of incoming packets into a simplified format of traffic signature, shown in Table 3, which is similar to a traffic signature.

Figure 6 Modules in the sentinel (see online version for colours)

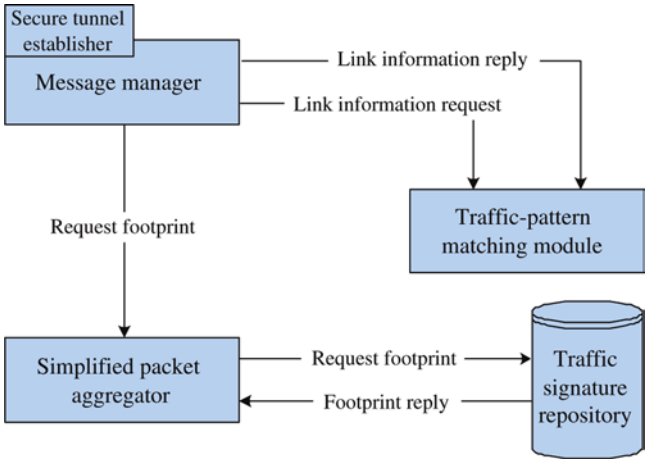


Table 3 Format of simplified traffic signature

Attributes	*Value*
1 No. of incoming TCP packets per t minute(s)	Numeric
2 No. of incoming UDP packets per t minute(s)	Numeric
3 No. of incoming ICMP packets per t minute(s)	Numeric

A sentinel collects the packets and transforms them into traffic signature. When receiving a traceback command, the message manager would identify the attack type in the command and perform the Grey Relational Analysis (GRA) to find the possible entrances of attack traffic.

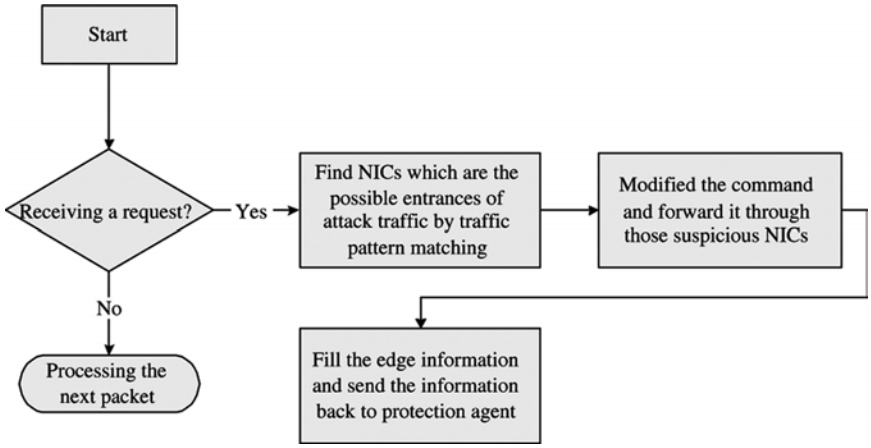
1 obtain the signature of the traffic flow

2 upon receiving a traceback command, a sentinel modifies and forwards the command to upstream sentinels that are the possible entrances of attack traffic identified by the traffic-pattern matching module

3 send the connection information back to the protection agent.

The flow chart of the whole procedure is depicted in Figure 7. According to the implementation results in Djalaliev et al. (2008), the sentinel greatly reduces

Figure 7 Operating flow chart in the sentinel (see online version for colours)



the impact of DDoS attacks on the response time. For more implementation details of the sentinel, the reader is referred to Djalaliev et al. (2008).

4 DDoS detection and traceback mechanism

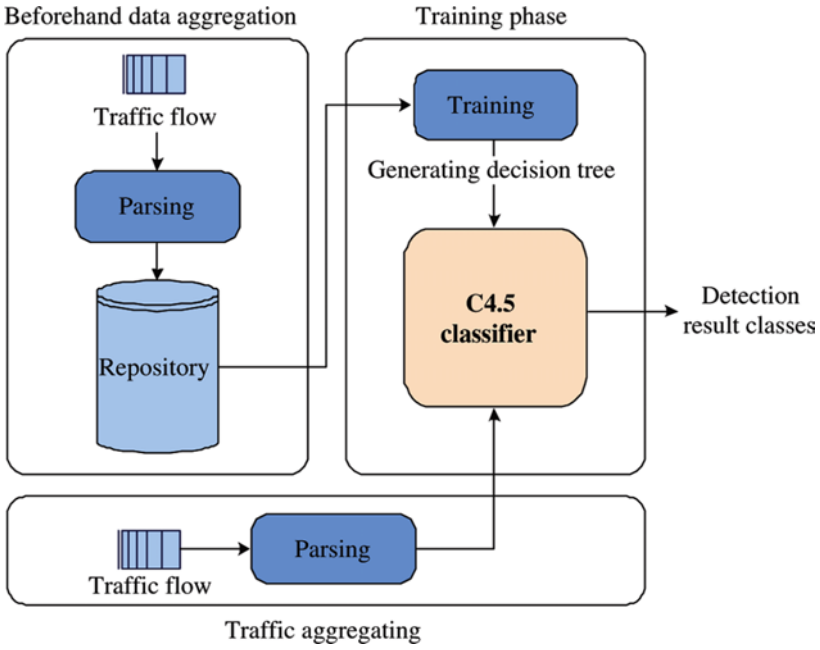
4.1 DDoS detection

It is reasonable to assume that the attack traffic would be different from the normal traffic in some aspects. We build a base-line traffic profile from the normal network traffic. Whenever the network traffic deviates from the base-line profile *significantly*, an attack is alarmed. We adopt a decision-tree classifier (Li and Chan, 2006; Rokach and Maimon, 2005) to classify network traffic. The advantage of the decision-tree classifier is its efficiency in both generalisation and new attack detection (Bouzida and Cuppens, 2006). A decision tree consists of leaf nodes representing classes and non-leaf nodes that specify tests to be carried out on a particular attribute. The construction of decision trees is based on training data. Then, the classifier is used to new data. The scenario of the proposed DDoS detection is illustrated in Figure 8.

We adopt the C4.5 (Quinlan, 1986, 1993) algorithm to construct the decision tree. C4.5 chooses the attribute as the splitting criterion according to the entropy-based gain ratio to overcome the over-fitting problem. First, C4.5 defines info(T) in equation (1). info(T) represents the entropy of the training data set T and represents the probability that one random instance from T belongs to a class C_j (there are four classes in our system: Normal, TCP SYN attack, UDP attack, and ICMP attack and one traffic signature aggregated per 1 min would be considered as one instance in our system).

$$\text{info}(T) = -\sum_{j=1}^{k} \left[\frac{|T_j|}{|T|} \times \log_2 \left(\frac{|T_j|}{|T|} \right) \right]. \quad (1)$$

Figure 8 DDoS detection scenario (see online version for colours)



Then, the gain information for an attribute is defined in equation (2). $\text{gain}(X)$ measures the quantity of information that is gained by partitioning T according to the attribute X (we treat the format of traffic signature defined in Table 2 as the attributes in our system).

$$\text{gain}(X) = \text{info}(T) - \sum_{i=1}^{n} \left[\frac{|T_i|}{|T|} \times \text{info}(T_i) \right] \tag{2}$$

where T_i represents the number of instances in the specific attribute. Then, the gain ratio is defined in equation (3).

$$\text{gain\_ratio}(X) = \frac{\text{gain}(X)}{-\sum_{i=1}^{n} \left[\frac{|T_i|}{|T|} \times \log_2\left(\frac{|T_i|}{|T|}\right) \right]}. \tag{3}$$

The attribute with the largest gain ratio is selected as the splitting criterion in the decision tree. On the basis of the selected attribute, the training data set is then divided into several subsets. Another attribute is similarly selected and each subset is further split. The splitting procedure is repeated until all the data in a subset belong to the same class or the gain ratios of all the attributes are the same. The construction procedure is summarised as follows:

1 Select the attribute with the largest gain ratio as the splitting criterion, and create a branch for each possible value of the selected attribute.

2 Divide the instances in the training data set into subsets according to the selected attribute.

3 Repeat Steps 1 and 2 for each branch.

In our implementation, we define four classes – normal, TCP SYN flooding, UDP flooding, ICMP flooding – and 16 attributes derived from the traffic signatures. A decision tree is then constructed from the training data set. According to the decision tree, the incoming traffic is classified.

4.2 Traceback

When the protection agent detects an attack, it raises an alarm to the traceback module. Because the source IP address in a packet could be easily spoofed, it cannot be used for traceback. Instead, we make use of traffic-flow pattern matching for traceback. Our objective is to find the routers where the attack traffic first enters the network. Starting from the victim, we attempt to discover the routers on the attack path one by one, until we reach the entry points of the attack traffic. For each router, we identify the incoming link on which the *incoming* traffic is most similar to the *outgoing* attack traffic. Then, that link is deemed to be on the attack path.

To determine the similarity of the traffic on the communication links, we make use of traffic-flow pattern matching (Kim and Helmy, 2005; Mansfield et al., 2000). There are two separate procedures in pattern matching: *trend-pattern matching* and *volume-pattern matching*. Traffic pattern matching is done in sentinels and the results are collected by the protection agent, which will construct the attack paths.

4.2.1 Attack edge determination

When a DDoS attack was detected, the traceback module will wait for sentinels to aggregate the traffic signatures. Then, the traceback module puts the attack traffic signatures aggregated during the attack and the timestamp when the attack was detected into the traceback command.

The traceback module in the protection agent issues a traceback command to the upstream sentinel, which is referred to as the *1-hop sentinel*. When the 1-hop sentinel receives the traceback command, it searches the traffic signature repository for every Network Interface Card (NIC) to retrieve the traffic signatures with the appropriate attack type and the aggregated timestamp that matches the timestamp in the traceback command. Afterwards, the sentinel applies the traffic-flow pattern-matching algorithm to identify the set of NICs that are the possible entrances the attack traffic may come from. If a router is equipped with n network interface cards, there will be $2^n - 2$ different combinations (the two cases – no NIC and all NICs – are ignored). And if $n = 1$, then this only one NIC will be considered as the entrance of attack traffic. After identifying the suspicious entrances of the attack traffic, the sentinel would send the connection information (which includes IP addresses of the two ends of the link and their distances from victim) to the protection agent through an SSH tunnel. In this way, the protection agent could receive the links that the attack traffic might pass through.

After sending back the connection information, the sentinel puts the traffic signatures aggregated by the suspicious NIC into the traceback command as new evidence and forwards this modified command to the upstream sentinel. When the upstream sentinel receives the traceback command, it repeats the same procedure until the entrance router is reached. Figure 9 illustrates the algorithm of the edge of attack path sampling. After all the connection information is collected, the attack paths could be reconstructed.

4.2.2 Traffic-flow pattern matching

In a DDoS attack, the attack traffic enters the network from multiple routers and flows to a single victim. The communication links that the attack traffic passes through form a tree (under normal routing) with the victim as the root and the entrances as the leaves. Our traceback method starts from the victim and identifies the routers on the tree one by one. Each sentinel will find the upstream routers on the tree.

The major problem in our traceback method lies in identifying the upstream routers of attack traffic in the sentinels. Therefore, the aim of traffic-flow pattern matching is to identify the subset of NICs on a router

Figure 9 Algorithm for determining the attack edges

```
1. begin Modified attack edge sampling:
2. find the set of NICs whose traffic is closest to the evidence
in the traceback command;
3. If command.ip == NULL then;
4.     fill the IP address of the suspicious NIC in the "ip"
field of the command;
5. else
6.     edge_info.start_ip := command_ip;
7.     edge_info.end_ip := ip address of the suspicious NIC;
8.     edge_info.distance := command.distance +1;
9.     send edge_info back to the protection agent;
10. endif
11. command.start := the ip address of the suspicious NIC;
12. increment the command.distance;
13. fill the traffic through the suspicious NICs in the "evi-
dence" field as new evidence;
14. forward the modified command to the suspicious NICs;
15. end
```

whose collective incoming traffic has similar signature as the attack traffic that goes out of that router. We apply two kinds of pattern-matching techniques, which measure both the *trend* and the *volume* of network traffic.

- *Trend-Pattern Matching*: Trend-pattern matching is based on the assumption that the DDoS attack traffic should dominate the *change* in the outgoing traffic from a router. Hence, we need to characterise the traffic trend quantitatively and determine if they are similar.

  Unlike other systems that compare traffic signatures with conventional statistical methods that favour the major sample space instead of small sample space and are easily affected by the extreme value, GRA (Deng, 1989; Lin and Liu, 2004) is used in our system. The GRA is applicable to a small sample size and could overcome the weakness of conventional statistic method by analysing the relation between factors from a small amount of data set. Because of the concern of the efficiency in traceback, the duration of the observation window[1] is quite short and the resulting sample size is quite small. Therefore, the GRA could satisfy our need and achieve the relational comparison in less effort within a large number of NICs. There are three pre-processing steps for the GRA:

  - grey relational maximising operation
  - grey relational coefficient computation
  - grey relational grade computation.

  During the observation window, the 1-hop sentinel computes a sequence of traffic signatures (one per minute) for each combination of NICs. The maximising operation (equation (4)) is then applied to normalise the signatures. The purpose of the maximising operation is to diminish the magnitude of sequences and make them comparable.

$$y(k) = \frac{x(k)}{x_{\max}} \tag{4}$$

  where the original signature sequence is $x = \{x(1), x(2), \ldots, x(n)\}$ and the normalised sequence is $y = \{y(1), y(2), \ldots, y(n)\}$.
  There is a sequence of signatures for each combination of NICs on a router. Among the sequences $y_1, y_2, \ldots$ etc., we wish to find the combination of NICs whose sequence of signatures is most similar to the sequence y_0 of the signatures of the attack traffic. We use equation (5) (Hsia et al., 2004) to compute grade $r(y_0, y_i)$, for each sequence y_i.

$$r(y_0, y_i) = \left(\frac{\Delta_{\max} - \overline{\Delta_{0i}}}{\Delta_{\max} - \Delta_{\min}} \right) \tag{5}$$

  where
  $\Delta_{\min} = \min_{\forall i} \min_{\forall k} \Delta_{0i}(k) = \min_{\forall i} \min_{\forall k} |y_0(k) - y_i(k)|$,
  $\Delta_{\max} = \max_{\forall i} \max_{\forall k} \Delta_{0i}(k) = \max_{\forall i} \max_{\forall k} |y_0(k) - y_i(k)|$
  and $\overline{\Delta_{0i}} = \frac{1}{n} \sum_{k=1}^{n} \Delta_{0i}(k)$
  $r(y_0, y_i)$ may be interpreted as the similarity between the sequences y_0 and y_i. If $r(y_0, y_1) > r(y_0, y_2)$, we may conclude that the sequence y_1 is more similar to the sequence y_0 than the sequence y_2.

- *Volume-pattern matching*: Trend-pattern matching considers only the similarity of two sequences but not their magnitude (here magnitude means the volume of traffic). *Volume-pattern matching* will compare the magnitudes of two sequences. We use equation (6) to compute a *volume coefficient* g_i for each sequence x_i.

$$g_i = \frac{\sum_{k=1}^{n} \sqrt{x_0(k) x_i(k)}}{\sum_{k=1}^{n} x_0(k)}. \tag{6}$$

4.2.3 Traceback command forwarding policy

The grade $r(y_0, y_i)$ represents the similarity in shape of the two sequences y_0 and y_i while the volume coefficient g_i represents the similarity in volume of the two sequences. When the grade $r(y_0, y_i)$ is greater than a selected threshold T_{trend}, we claim that the two sequences have the same shape. Similarly, when the volume coefficient g_i is greater than a certain threshold T_{vol}, we claim that two sequences have the same volume. In our system, we use $T_{\text{trend}} = 0.8$ and $T_{\text{vol}} = 0.9$. This would reduce the false positive/negative ratios in our experiment.

We consider only the sequences x_i for which $r(y_0, y_i) > T_{\text{trend}}$ and $g_i > T_{\text{vol}}$. Among these sequences, we choose the sequence with the largest $r(y_0, y_i)$. The subsets of NICs corresponding to the chosen sequence are deemed as the entrances for the attack traffic to enter

the router. When there is no sequence for which $g_i > T_{\text{vol}}$, we claim that all the NICs are the entrances for the attack traffic to enter the router. Otherwise, the router is not on the attack path and the sentinel on the router will stop forwarding the traceback command.

5 Experiment design and results

5.1 Simulation design

We verified the performance of the proposed detection and traceback system on the DETER test-bed (Benzel et al., 2006). The DETER test-bed provides users an environment to emulate the real-world network traffic with an easy-to-use web interface and various tools, such as SEER (Schwab et al., 2007), a benchmark for DDoS defence mechanism. There are three major components in a DDoS attack experiment: topology design, legitimate traffic (background traffic) and attack traffic.

5.1.1 Topology design

In our experiment, there are 5 zombie attackers, 20 routers and 10 clients, which perform common web browsing. The network topology is generated with the Waxman algorithm (Waxman, 1988), which is shown in Figure 10.

5.1.2 Legitimate traffic generation

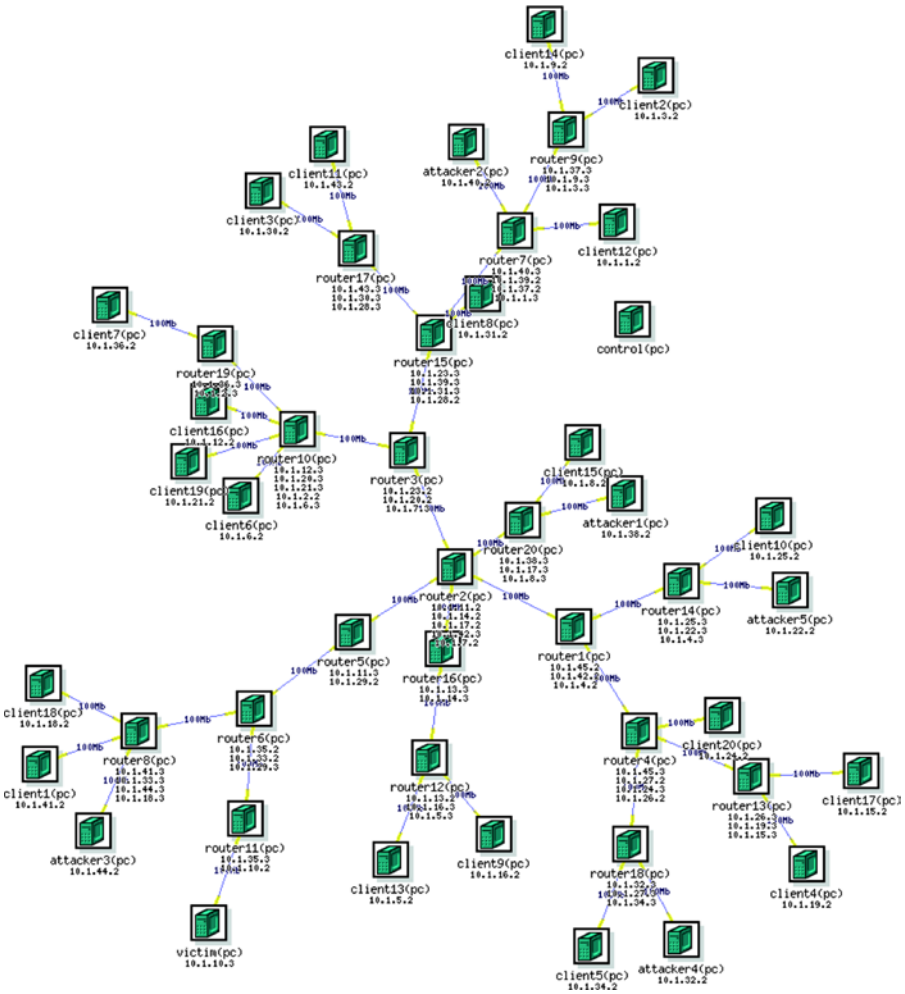
The background traffic (i.e., normal traffic, without attacks) is generated with harpoon (Sommers et al., 2004) from the actual trace data collected at the computing centre of Department of Computer Science in National Chiao Tung University. The machine is a web page server in the centre.

5.1.3 Experiment scenarios

The background traffic was collected from 25 June, 2008 (Wednesday) midnight to 27 June, 2008 (Friday) midnights (48 h in total), which is divided into two groups: the traffic on 25 June (denoted as *data25*) is used as training data set while the traffic collected on 26 June (denoted as *data26*) for testing purpose. Each day is further divided into four periods: bed-time (0 am–7 am), morning (7 am–12 pm), afternoon (12 pm–18 pm) and evening (18 pm–24 am).

There are four iterations in our experiment. The first two iterations constitute the training phase while the

Figure 10 Experiment topology design (see online version for colours)



last two the testing phase. First, we feed the simulation environment with data25. We obtain 1440 signatures (one per minute) for the background traffic. Second, we feed the simulation with data25 plus randomly generated attack traffic. We obtain another 1440 signatures and these signatures with attack traffic would be denoted as attack traffic signatures. The two sets of signatures are used to build the decision tree with the C4.5 algorithm. Third, we feed the simulation with data26. The resulting 1440 signatures are used to calculate the false positive ratio. Finally, we feed the simulation with data26 and randomly generated attack traffic. The resulting 1440 signatures will be used to calculate the false negative ratio and the false classification ratio.

5.1.4 Attack traffic generation

The attack traffic is randomly generated with the SEER tool. We tested three kinds of attacks: TCP SYN flood, UDP flood and ICMP flood. To simplify the experiment, at most one attack is underway at any time. Each attack lasts for 1 h in the training phase and for 12 min during the testing phase. The amount of attack traffic for each attack during the training phase is shown in Table 4. The amount of attack traffic during the testing phase is shown in Table 5. The testing phase is repeated three times, each with different attack traffic.

Table 4 Attack scenario in training data

TCP SYN flood	*UDP flood*	*ICMP flood*
150 pkt/per sec Pkt. size: 66 bytes	150 pkt/per sec Pkt. size: 256 bytes	150 pkt/per sec Pkt. size: 256 bytes

Table 5 Attack scenario for evaluating purpose

TCP SYN flood	*UDP flood*	*ICMP flood*
Scenario 1		
250 pkt/per sec Pkt. size: 66 bytes	250 pkt/per sec Pkt. size: 256 bytes	250 pkt/per sec Pkt. size: 256 bytes
Scenario 2		
150 pkt/per sec Pkt. size: 66 bytes	150 pkt/per sec Pkt. size: 256 bytes	150 pkt/per sec Pkt. size: 256 bytes
Scenario 3		
70 pkt/per sec Pkt. size: 66 bytes	70 pkt/per sec Pkt. size: 256 bytes	70 pkt/per sec Pkt. size: 256 bytes

5.2 Performance evaluation

5.2.1 Performance metric

- *Performance metrics for DDoS detection*: In detecting DDoS attacks, we focus on four metrics: (FNR False Negative Ratio), (FPR False Positive Ratio), (FCR False Classification Ratio) and detection latency. According to Table 6, the definitions of FNR and FPR are listed as equations (7) and (8), respectively.

Table 6 Situation analysis in detection

	Detection result	
Actual situation	*Attack*	*Normal*
Attack	A	B
Normal	C	D

'A' is the number of attack signatures that are successfully and correctly detected by the protection agent; 'B' is the number of attack signatures that the protection agent failed to detect; 'C' is the number of reported attack signatures while there is actually no attack; and 'D' is the number of normal traffic signatures that are recognised as normal (that is, not identified as an attack).

Table 7 defines the FCR for different attacks. In TCP SYN flooding attack, the false classification represents the ratio between the number of TCP SYN flooding attack traffic signatures that are detected as UDP flooding attack signatures or ICMP flooding attack signatures and the total number of TCP SYN flooding attack traffic signatures. The false classification ratio in UDP flooding attack and ICMP flooding are deduced in the same way. Detection latency represents the average number of time slots needed to detect the attack when the attack was launched in our experiment.

$$\mathrm{FNR} = \frac{B}{A+B} \tag{7}$$

$$\mathrm{FPR} = \frac{C}{C+D}. \tag{8}$$

Table 7 The false classification ratio

	Equation
TCP SYN flood	$\frac{N_{i=ICMP}+N_{i=UDP}}{R_{j=TCP}}$
UDP flood	$\frac{N_{i=ICMP}+N_{i=TCP}}{R_{j=UDP}}$
ICMP flood	$\frac{N_{i=TCP}+N_{i=UDP}}{R_{j=ICMP}}$

N_i represents the number of attack traffic signatures that the protection agent detected as attack i; R_j represents the number of attack traffic signatures actually generated by attack j.

According to Table 6, the FNR (see equation (7)) represents the ratio between the number of attack traffic signatures that are not detected and the total number of attack traffic signatures. FPR (see equation (8)) represents the ratio between the number of normal traffic signatures that are claimed as attack signatures and the total number of normal signatures.

- *Performance metrics for DDoS traceback*: In DDoS traceback performance evaluation, we define the Misidentified Normal Edge Ratio (MNER) and Misidentified Attack Edge Ratio (MAER) as our metrics for traceback. According to Table 8, MNER (see equation (9)) represents the ratio between the numbers of normal edges but are claimed as attack edges by the sentinels

and the total number of normal edges. MAER (see equation (10)) represents the ratio between the number of attack edges that are not found by the sentinels and the total number of attack edges.

$$\text{MNER} = \frac{G}{G+H} \tag{9}$$

$$\text{MAER} = \frac{F}{E+F}. \tag{10}$$

Table 8 Situation analysis in traceback

Actual situation	*Report result*	
	Attack path	*Normal path*
Attack path	e	f
Normal path	g	h

'e' is the number of attack edges that are successfully and correctly reported by sentinels; 'f' is the number of attack edges the sentinels failed to identify; 'g' is the number of edges that are not on the attack path but are mistakenly identified as attack edges by the sentinels and 'h' is the total number of edges that are not on the attack path and are recognised as normal.

5.2.2 *Simulation results*

- *Performance of DDoS detection*: Figure 11 depicts the false positive ratios for the four periods in a day. The result indicates that the false positive ratio ranges from 1.2% (bed time) to 2.4% (morning). In Noh et al. (2003), the false positive ratio ranges from 1% to 8% depending on the background traffic. However, it is not clear the amount of attack traffic in Noh et al. (2003). In D-WARD (Mirkovic and Reiher, 2005), the false positive ratio (which is called *false alarm*) is about 2%. However, it is clear about the amount of attack and normal traffic.

Figures 12–14 depict the false negative ratios of TCP SYN flooding, UDP flooding, and ICMP flooding, respectively. Because the sending rates in the ICMP flooding attack and the UDP flooding attack are the same, the results in ICMP and UDP flooding attacks are similar. When the attack rate is 150 packets per second (note that in the training phase the attack rate is 150 packets per second), the false negative ratio ranges from 5–10% for UDP and ICMP flooding. The false negative ratio is 2–3% for the TCP SYN flooding.

Figures 15–17 depict the results in the false classification ratios. The results show that the false classification ratio for the TCP SYN attacks is lower than that for ICMP and UDP flooding attacks. Nearly 40–50% of ICMP attacks may be mistaken as UDP attacks. Similarly, nearly 40–50% of UDP attacks may be mistaken as ICMP attacks. On the other hand, TCP SYN attacks are seldom mis-classified.

Another important issue in DDoS detection is the *detection latency*, i.e., how soon the system will claim an attack after the attack traffic reaches the victim. In our system, time is sliced into 1-minute slots. According to the results in Figures 18–20, our system could claim an attack within 1–1.4 min under different attack rates.

- *Performance of attacker traceback*: When reconstructing the attack paths, it is possible to mistake an edge that is *not* on the attack path as an

Figure 11 False positive ratio in DDoS detection (see online version for colours)

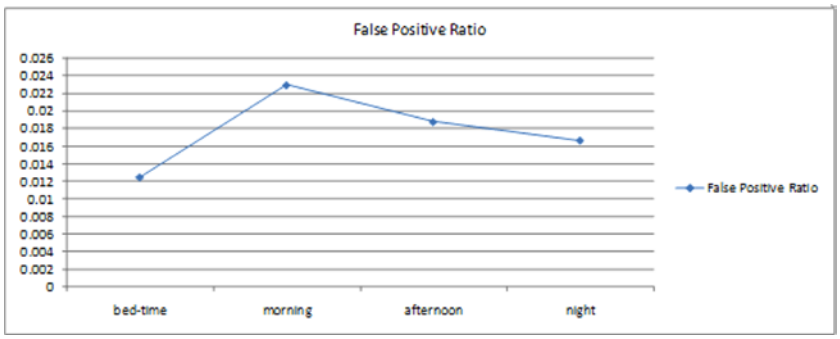


Figure 12 False Negative Ratio in TCP SYN flooding (see online version for colours)

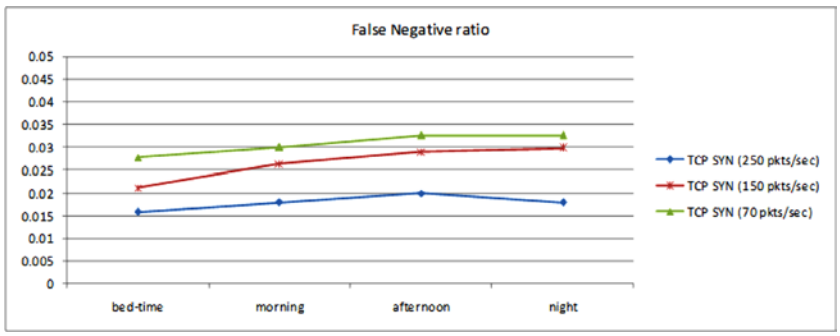


Figure 13 False Negative Ratio in UDP flooding (see online version for colours)

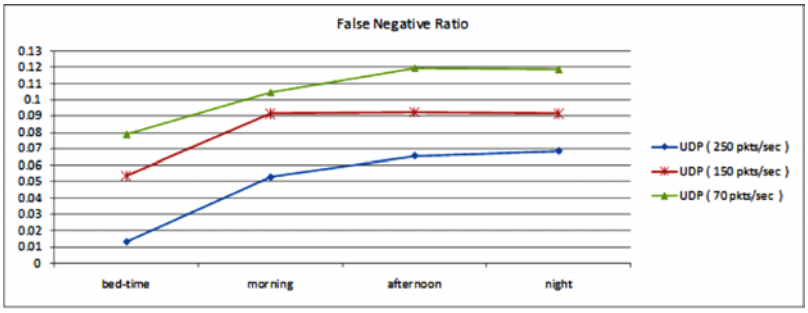


Figure 14 False Negative Ratio in ICMP flooding (see online version for colours)

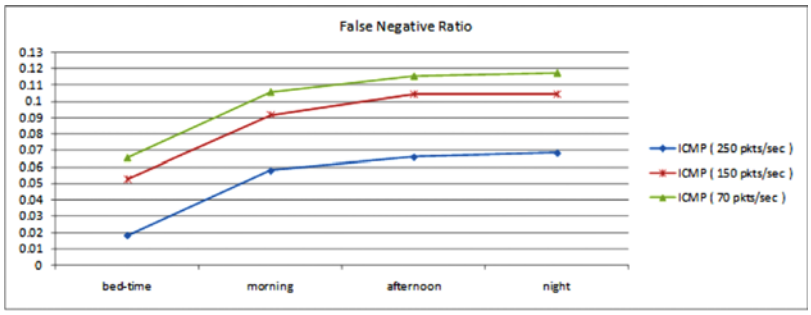


Figure 15 False Classification Ratio in TCP SYN flooding (see online version for colours)

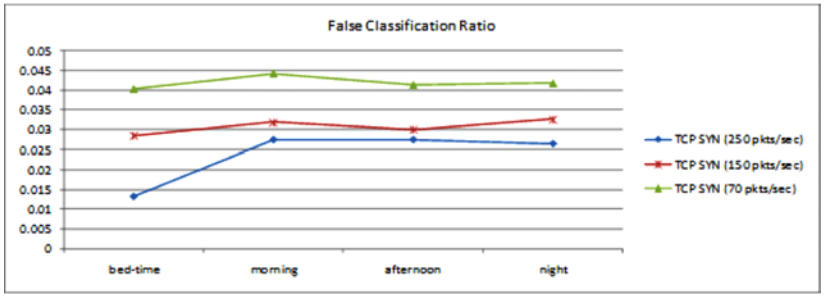


Figure 16 False Classification Ratio in UDP flooding (see online version for colours)

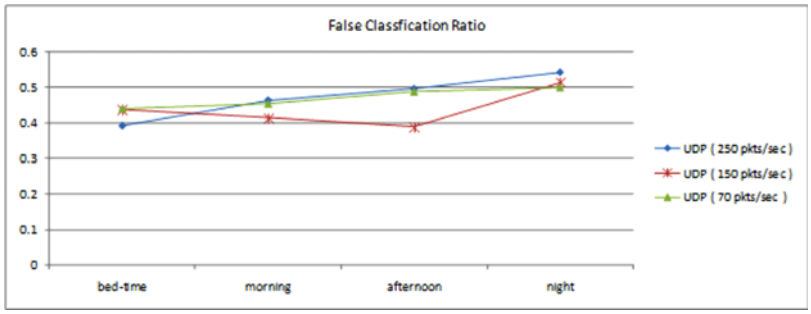


Figure 17 False Classification Ratio in ICMP flooding (see online version for colours)

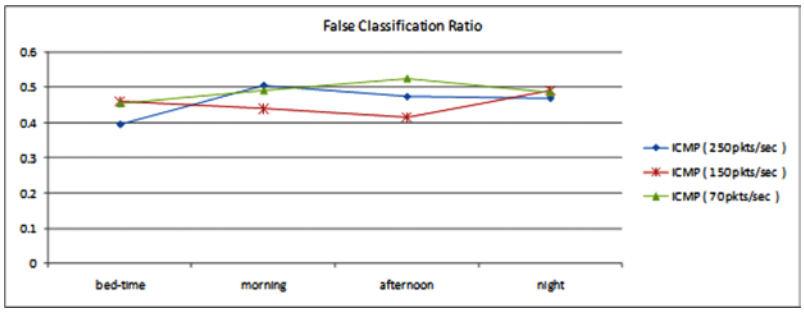


Figure 18 Detection latency in TCP SYN flooding (see online version for colours)

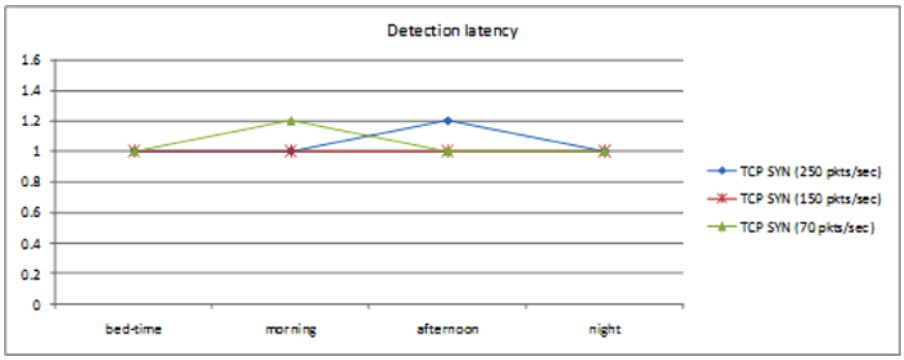


Figure 19 Detection latency in UDP flooding (see online version for colours)

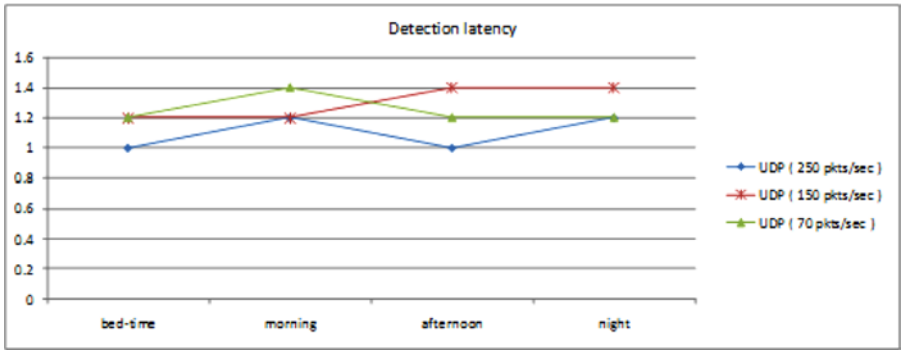
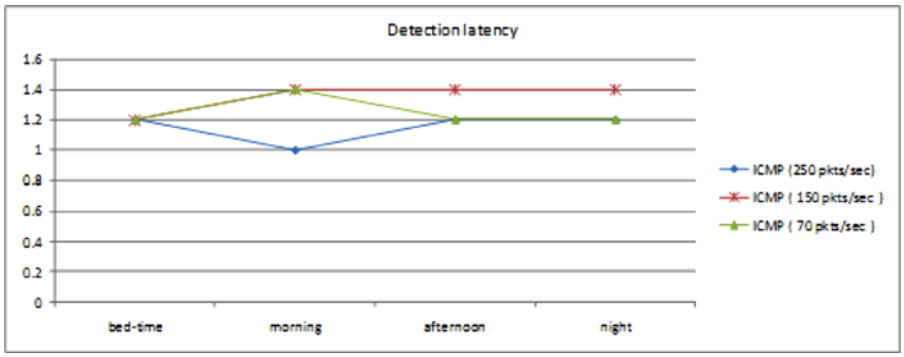


Figure 20 Detection latency in ICMP flooding (see online version for colours)



attack edge and vice versa. Figures 21 and 22 show MNER and MAER with different observed windows and different trend-pattern thresholds. Remember the *observed window* is the amount of time the sentinels collect traffic data after an attack is claimed. Figure 22 shows that MAER is almost a constant while $T_{\text{trend}} < 0.9$ regardless of the observed window. The results also verify that the GRA is suitable for small sample space (size of sample space < 30).

Figure 21 Misidentified Normal Edge Ratio (MNER) in traceback (see online version for colours)

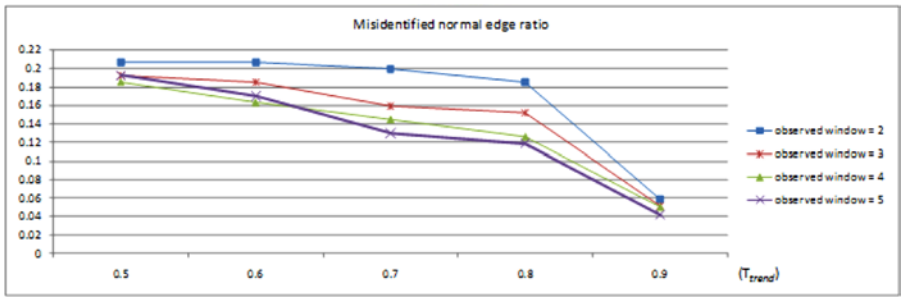
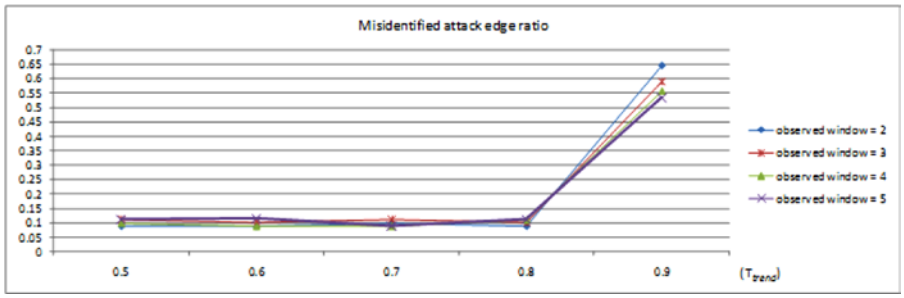


Figure 22 Misidentified Attack Edge Ratio (MAER) in traceback (see online version for colours)



When we keep MAER low (i.e., the $T_{\text{trend}} < 0.9$), the lowest MNER is around 12–18.5% (from Figure 21). Furthermore, we enforce an observed window for at least 3 min, MNER falls between 12 and 14%. In Izaddoost et al. (2007), MNER is 17–19% in old *iTrace* model and 6% in new proposed model under different network traffic. MNER in Figure 21 is less than 10% but that system makes use of a modified probability packet marking mechanism, which involves many other issues.

6 Conclusions

In this paper, we propose a DDoS defence system, which includes attack detection by decision tree and attacker traceback with traffic-pattern matching. Our system is based on the observation that the network traffic under DDoS attack would differ from the traffic in normal situation. We apply the decision tree (C4.5) generating algorithm to construct the classification model and detect abnormal traffic flow. In traceback phase, we use a novel traffic pattern-matching procedure, which is based on GRA, to identify the traffic flow that is similar to the attack flow and, based on this similarity, to trace back the origin of an attack. The attack path reconstruction is then accomplished by the protection agent and the sentinels.

We conduct our experiment on the DETER system. According to our experiment results, our system could detect the DDoS attack with the false positive ratio about 1.2–2.4%, false negative ratio about 2–10% with different attacks and attack sending rates and find the attack path in traceback. The misidentified attack edge ratio is about 8–12% and misidentified normal edge ratio about 12–14%. The result indicates that our proposed system is capable of detecting the attacks and tracing them back with high accuracy and within a short time.

Acknowledgement

The work reported in this paper is partially supported by National Science Council (NSC), Taiwan, Republic of China, under grants NSC 96-2628-E-009-014-MY3, NSC 97-2218-E-009-029, NSC 97-2623-7-036-001-D, NSC 97-2221-E-009-048-MY3 and NSC 97-2221-E-009-049-MY3.

References

Bellovin, S.M., Leech, M. and Taylor, T. (2001) *ICMP Traceback Messages*, IETF, Internet Draft.

Benzel, T., Braden, R., Kim, D., Neuman, C., Joseph, A. and Sklower, K. (2006) 'Experience with DETER: a testbed for security research', *Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM'06)*, March.

Berral, J.L., Poggi, N., Alonso, J., Gavaldà, R., Torres, J. and Parashar, M. (2008) 'Adaptive distributed mechanism against flooding network attacks based on machine learning', *Proceedings of the 1st ACM Workshop on Workshop on AISec*, October, pp.43–50.

Bloom, B H. (1970) 'Space/time trade-offs in hash coding with allowable errors', *Communications of the ACM*, Vol. 13, No. 7, July, pp.422–426.

Bouzida, Y. and Cuppens, F. (2006) 'Neural networks vs. decision trees for intrusion detection', *IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM'06)*, September.

Burch, H. (2000) 'Tracing anonymous packets to their approximate source', *Proceedings of the 14th USENIX Conference on System Administration*, December, pp.319–328.

Deng, J.L. (1989) 'Introduction to grey system theory', *The Journal of Grey System*, Vol. 1, No. 1, November, pp.1–24.

Djalaliev, P., Jamshed, M., Farnan, N. and Brustoloni, J. (2008) 'Sentinel: hardware-accelerated mitigation of bot-based DDoS attacks', *Proceedings of the 17th International Conference on Computer Communications and Networks*, August, pp.633–640.

Douligeris, C. and Mitrokotsa, A. (2004) 'DDoS attacks and defense mechanisms: classification and state-of-the-art', *Computer Networks*, Vol. 44, No. 5, April, pp.643–666.

Fallah, M.S. (2010) 'A puzzle-based defense strategy against flooding attacks using game theory', *IEEE Transactions on Dependable and Secure Computing*, Vol. 7, No. 1, January–March, pp.5–19.

Hsia, K.H., Chen, M.Y. and Chang, M.C. (2004) 'Comments on data pre-processing for grey relational analysis', *Journal of Grey System*, Vol. 7, No. 1, June, pp.15–20.

Izaddoost, A. Othman, M. and Rasid, M.F.A. (2007) 'Accurate ICMP traceback model under DoS/DDoS attack', *Proceedings of the International Conference on Advanced Computing and Communications (ADCOM'07)*, December, pp.441–446.

Kim, Y. and Helmy, A. (2005) 'SWAT: small world-based attacker traceback in ad-hoc networks', *Proceedings of the 2nd International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'05)*, July, pp.85–96.

Li, X. and Chan, C.W. (2006) 'Applying a machine intelligence algorithm for prediction', *Proceedings of the International Conference on Computational Intelligence and Security*, Vol. 1, November, pp.793–796.

Lim B. and Uddin, M.S. (2005) 'Statistical-based SYN-flooding detection using programmable network processor', *Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA'05)*, Vol. 2, July, pp.465–470.

Lin, Y. and Liu, S. (2004) 'A historical introduction to grey systems theory', *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 3, October, pp.2403–2408.

Mansfield, G., Ohta, K., Takei, Y., Kato, N. and Nemoto, Y. (2000) 'Towards trapping wily intruders in the large', *Computer Networks*, Vol. 34, No. 4, October, pp.659–670.

Mirkovic J. and Reiher, P. (2005) 'D-WARD: a source-end defense against flooding denial-of-service attacks', *IEEE Transactions on Dependable and Secure Computing*, Vol. 2, No. 3, July–September, pp.216–232.

Mirkovic, J. and Reiher, P. (2004) 'A taxonomy of DDoS attack and DDoS defense mechanisms', *ACM SIGCOMM Computer Communications Review*, Vol. 34, No. 2, April, pp.39–54.

Mirkovic, J., Prier, G. and Reiher, P. (2003) 'Source-end DDoS defense', *Proceedings of the 2nd IEEE International Symposium on Network Computing and Applications (NCA'03)*, April, pp.171–178.

Noh, S., Lee, C., Choi, K. and Jung, G. (2003) 'Detecting distributed denial of service (DDoS) attacks through inductive learning', *Proceedings of the 4th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'03)*, March, pp.286–295.

Öke, G. and Loukas, G. (2007) 'A denial of service detector based on maximum likelihood detection and the random neural network', *The Computer Journal*, Vol. 50, No. 6, September, pp.717–727.

Peng, T., Leckie, C. and Ramamohanarao, K. (2007) 'Survey of network-based defense mechanisms countering the DoS and DDoS problems', *ACM Computing Surveys*, Vol. 39, No. 1, pp.1–45.

Qu, H., Su, P., Lin, D. and Feng, D. (2005) 'A packet marking scheme for IP traceback', *Proceedings of the International Conference on Networking*, April, pp.964–971.

Quinlan, J.R. (1986) 'Induction of decision trees', *Machine Learning*, Vol. 1, No. 1, pp.81–106.

Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA.

Rokach, L. and Maimon, O. (2005) 'Top-down induction of decision trees classifier – a survey', *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 35, No. 4, November, pp.476–487.

Savage, S., Wetherall, D., Karlin, A. and Anderson, T. (2000) 'Practical network support for IP traceback', *ACM SIGCOMM Computer Communications Review*, Vol. 30, No. 4, August, pp.295–306.

Schwab, S., Wilson, B., Ko, C. and Hussain, A. (2007) 'SEER: a security experimentation environment for DETER', *Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test*, August.

Sommers, J., Kim, H. and Barford, P. (2004) 'Harpoon: a flow-level traffic generator for router and network tests', *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, June, pp.392–393.

Stone, R. (2000) 'CenterTrack: an IP overlay network for tracking DoS floods', *Proceedings of the 9th Conference on USENIX Security Symposium*, Vol. 9, July, pp.15.

Waxman, B.M. (1988) 'Routing of multipoint connections', *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, December, pp.1617–1622.

Zaroo, P. (2002) *A survey of DDoS Attacks and Some DDoS Defense Mechanisms, Technical Report, Lecture Notes for Advanced Information Assurance (CS626)*, Computer Science, Purdue University.

Website

Tcpdump/libpcap, http://www.tcpdump.org/