Detection of the distributed denial of service attack

Dai Azami s1070004 Supervised by Hiroshi Toyoizumi

Abstract

We research about the effect of a DoS attack. In order to detect a concrete effect, we built the program which draws waiting time from the difference of two RTT sent simultaneously. The program was used and the effect of the attack by the DoS tool and the script attack using the browser was detected. Moreover, we experimented to the server include IDS with the function which prevents a DOS attack. Consequently IDS was become clear that was effective in the DoS attack although, there is no effect in DDoS.

1 Introduction

For several years, the computer network technology has developed rapidly. The means of hacking is also included in it.

Now, there are means of hacking. (D)DoS ((distributed) denial of service) attack is one of the strong computer attack. DoS attack is that the attacker sends large amount of data for target server(See left of Figure 1). Them server performance decline, and clients are forced to more wait. In consequence, DoS attack block the service which a server offers (ex. WWW, FTP, DNS, and SNTP). If target server can not endure the attack, it stop the service. Mainly, DoS attack do using tool.
DDoS attack is means that DoS attack by the plural attackers(See right of Figure 1). This attack send data smaller than DoS attack. However, the more attackers increase, the effect of DDoS attack increase. CAJ [1] is warning of the serious damage which DDoS brings about. Since plural attacker are required, DDoS attack is used well as a function of a computer virus.

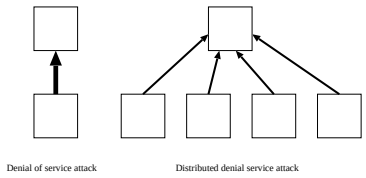

Figure 1: The means of DoS and DDoS

There are a lot of service which receives a DOS attack. We chose HTTP servise as the object of detection. Because, it is the most popular service, and it is tended to aim at it (D)DoS attack. In fact, the DOS attack on HTTP service is very easy, and it become a DOS attack only by repeatedly hitting the relord button of a web browser. And, we referred to Mr. Mizutani's thesis [3] about the offensive detection method. He [3] announced the method of detecting waiting time from RTT (round trip time) of a PING signal. Therefore, we also propose a method of detecting waiting time from RTT of a HTTP/GET signal.

2 Experiment

The contents of the experiment are as follows. It can be performed even if there is no skill of hacking.

2.1 Round Trip Time

RTT is defined as time until a packet reaches a host and returns from a client in the network. And, the time while a host is responding is also contained. (See Figure 2) Since RTT is influenced by traffic, if the server has attacked by DoS attack, the network traffic increases and RTT increases also. That is, the effect of a DoS attack can be measured by change of RTT. In the HTTP proto-

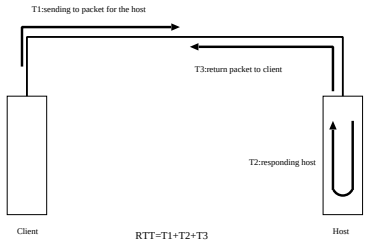

Figure 2: The definition of RTT

col, RTT has pointed out the following time.

1. Client sends HTTP/GET signal for server
2. Signal waits for the server to accept
3. Server receives request signal
4. Server sends data to client
5. Client receives data

Therefore, the effect can be measured by a tool which observes RTT.

2.2 The Calculate the waiting time from RTT

Actually, in order to know an attack effect, we have to detect service waiting time from RTT. Therefore, two signals are sent simultaneously and the difference of RTT is asked for waiting time. Client sends two request signals to the Server at the same time, and this operation is repeated several times. If these 'i'th two RTT are set to $RTT_1[i]$, $RTT_2[i]$, the signals set time to go to and for in a network top to N[i] (N[i]=na+nb), the time which is waiting for service is w[i], and the time which has received service is set to S[i] (See Figure 3).

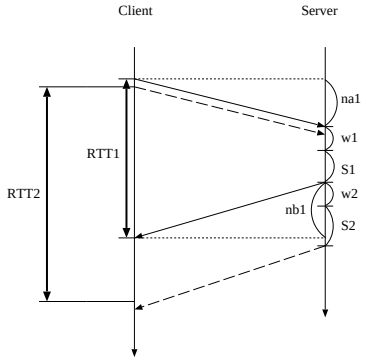

Figure 3: The flow of data and RTT

$$RTT_1[i] = N_1[i] + w_1[i] + S_1. \quad (1)$$

$$RTT_2[i] = N_2[i] + w_1[i] + S_1 + w_2[i] + S_2. \quad (2)$$

Now, both have left simultaneously. That is,

$$N_1[i] = N_2[i].$$

Moreover, both services are the same.

$$S = S_1 = S_2.$$

That is, the (1) and (2) are

$$RTT_1[i] = N[i] + w_1[i] + S.$$

$$RTT_2[i] = N[i] + w_1[i] + w_2[i] + 2S.$$

and therefore,

$$RTT_2[i] - RTT_1[i] = w_2[i] + S.$$

If $RTT_2[i] - RTT_1[i]$ sets the smallest value to RTT_{min},

$$RTT_{min} = w_2 + S.$$

S is assumed to be constant.

$$w_2 = 0.$$

$$RTT_{min} = S.$$

In this way, the value of S becomes settled and can measure the value of changing w correctly. Strictly speaking, this w is not waiting time of RTT2, because the waiting time and service time of the last signal is contained in this. However, w is defined as waiting time expediently.

2.3 Algorithm

We create the following algorithm using this theory.

1. transmit two HTTP/GET signals to a server at once
2. detect the time lag of the two replies
3. repeat several times, and find provisional service time from the minimum value
4. measure waiting time from the provisional service time
5. if waiting time becomes a negative value, a provisional service period will be re-set up

We created this tool program by C.

3 Experimental result

We experimented by using waiting time measurement tool. There result are given below.

3.1 Result of experiment 1

In order to confirm the validity of the tool, We measured the waiting time of a certain web site for about 20 hours. However, We set up the interval of measurement in 5 minutes so that service might not be interrupted.

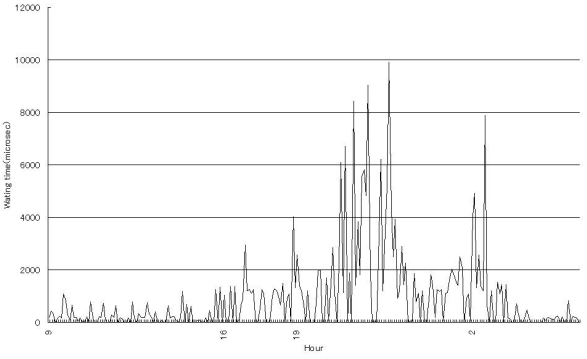

Figure 4: Waiting time measurement

The Figure 4 points out that the average waiting time increases on term of midnight, that is, the network traffic increase on this term. It was verified by this experiment that this tool can detect time.

3.2 Result of experiment 2

In order to investigate the effect of a (D)DoS attack, the change of waiting time was measured using the attack tool in private network. We did two experiment. One is a DoS attack tool (phack3w). This tool sends a lot of GET signals automatically toward a server. The other is a multiplex access script (T cannon script) using a browser (which has been used as a means of DDoS in the past). This script makes a browser repeat reloading, and sends a GET signal indirectly.

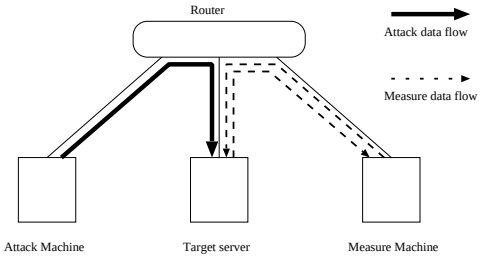

Figure 5: Arrangement of each machine

	Normal	DoS tool	script attack
Average	28.9	3038027	4057

Table 1: Average of waiting time (Ex. 2)

The result of the waiting time is shown in Figure 6. A vertical axis shows waiting time and the horizontal axis shows experiment time, and it attacked between A and B. The table 1 expresses each average waiting time. The attack by the DoS attack tool had the very large effect, and had great influence on waiting time. The script attack had some effects. However, since it is very small compared with a DoS tool, it has not appeared in graph(See bottom graph of Figure 6).

3.3 Result of experiment 3

There are some means of defense. IDS is one of it. IDS has the function of the packet filtering. This function is that reject the barbed packet. In this case, IDS regards repeated access that come from DoS attack as barbed packet, and reject them. Next experiment is that attack the machine which installed IDS. As IDS for an experiment in us Zone Alerm [2] was used. The attack method is the same as in experiment 2 (see figure 7).

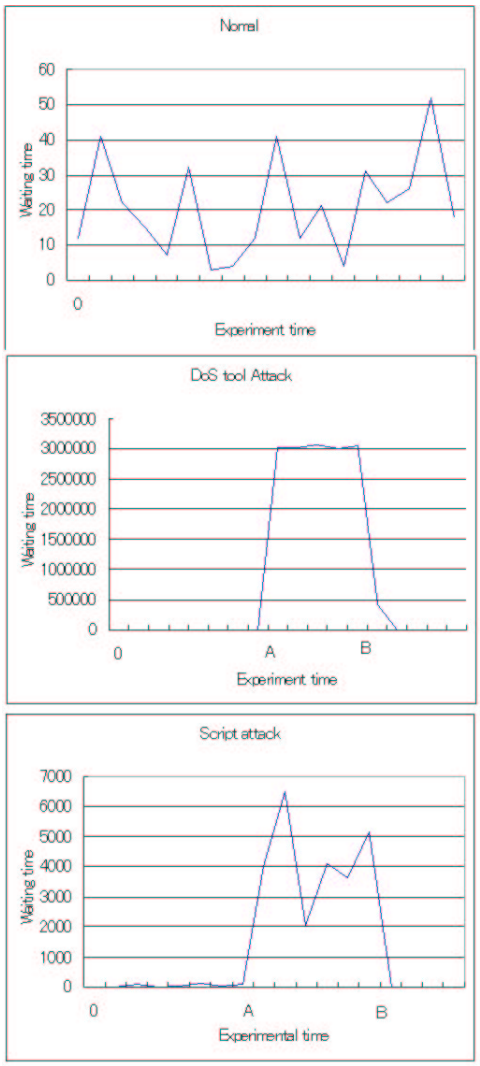

Figure 6: The effect of attack

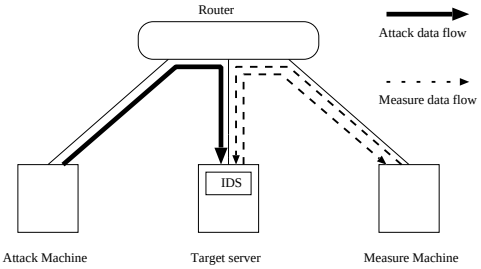

Figure 7: Arrangement of each machine

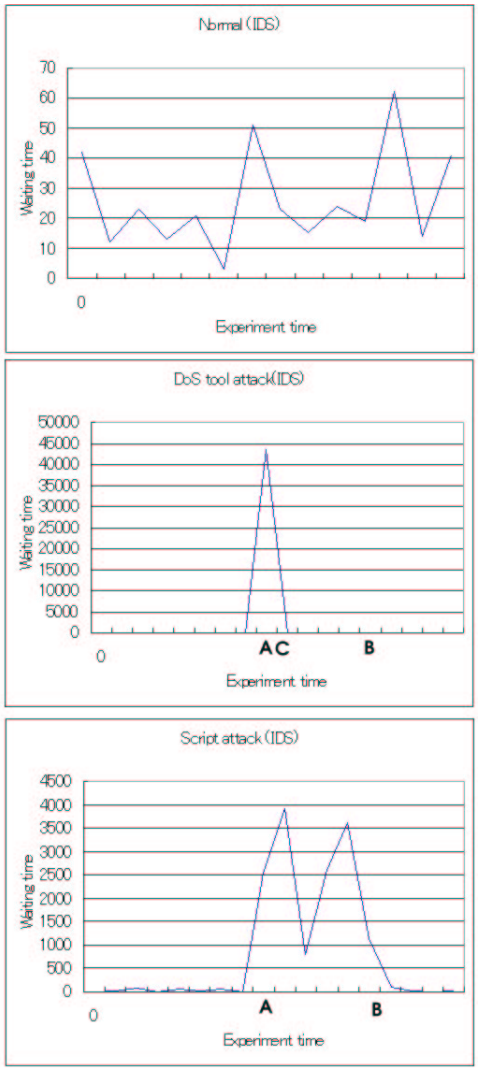

Figure 8: The effect of attack (include IDS)

	Normal	DoS tool	script attack
Average	28.3	52.2	2738

Table 2: Average of waiting time (Ex. 3)

The result of the waiting time is shown in Figure 8. In the time C immediately after a DOS attack start, IDS has detected the DoS. And, it performed filtering. Since the attack was intercepted by IDS, the waiting time of a DoS attack was recovered immediately. This indicate that the IDS is effective way. However, IDS could not detect the script attack, and waiting time did not big effect either.

4 Conclusion

It was verified by experiment that is possible to ask two RTT for waiting time. From the result of the experiment using this tool, we have detected very large effect by DoS tool attack. Similarly, we have detected the small effect by script attack. From the result of the experiment include IDS, the DoS tool attack was detected by IDS and prevented by packet filtering function. However, IDS could not detect the script attack. DoS attack has a big effect. However, it is easy to detect and diffence it. Thereto script attack has a small effect by one machine. but it is difficult to detect it. Therefore, defense is almost impossible.

4.1 Future works

We experimented in (D)DoS attack on privete network. Consequently, we have detected these attacks. However, the actual attack is delivered in a public network. We want to experiment in (D)DoS attack to a server on the public network.

Acknowledgement

I want to thank Mr. Mizutani for his research. His theory was helpful to lead waiting time. I also thank Mr. Maeda for his cooperation in the experiments.

References

[1] C. Assosiates, “Virus infomation center (ddos),” http://www.caj.co.jp/virusinfo/2002/ddos.htm, 2002.

[2] Z. Labs, “Zone alarm,” http://www.zonelabs.com/store/content/home.jsp.

[3] N. Mizutani, Detection of Hacking by Observing Traffic on Network, Ph.D. thesis, University of Aizu, 2001.