International Journal of Cyber-Security and Digital Forensics (IJCSDF) 4(2): 339-353
1The Society of Digital Information and Wireless Communications, 2015 (ISSN: 2305-0012)

Analysis of Slow Read DoS Attack and Countermeasures on Web servers

Junhan Park, Keisuke Iwai, Hidema Tanaka and Takakazu Kurokawa
National Defense Academy of Japan
1-10-20 Hashirimizu, Yokosuka-Shi, Kanagawa-Ken, 239 -8686, Japan
junhanp78@gmail.com, {iwai, hidema, kuro}@nda.ac.jp

ABSTRACT

The ideas and techniques of DoS (Denial of Service) and DDoS (Distributed DoS) Attack strategies become more effective and more complex. In this paper, we focus on a Slow Read DoS Attack which is one of the sophisticated DoS attack techniques. This technique prolongs time to read the response from the Web server, although an attacker sends a legitimate HTTP request. When an attacker sends many such legitimate requests, he can keep many open connections to Web server and eventually cause DoS situation. In this paper, we analyze the effectiveness of Slow Read DoS Attack using the virtual network environment. As the result, we can find that Slow Read DoS Attack by a single attacker can be prevented by adequate security settings of Web server and applying countermeasure such as ModSecurity. However, from the analysis of Slow Read DoS Attack technique, we can also find that these countermeasures are not effective against distributed Slow Read DoS Attack (Slow Read Distributed DoS Attack) which is proposed in this paper.

KEYWORDS

Slow Read DoS Attack, Web Server Security, Slowhttptest, Apache, ModSecurity

1 INTRODUCTION

DoS (Denial of Service) attacks are evolved and consolidated as severe security threats to network service. Earlier DoS attacks use flood-based high-bandwidth approach and exploit the resource of network and transport protocol layers. Since the strategies of DoS attacks are simple, they can be prevented by filtering the source IP address. In order to break through this countermeasure, DDoS (Distributed DoS) attacks deliver a DoS attack by many attackers. However, there is a problem using high-bandwidth in DoS and DDoS attacks, and many improvements are proposed. One of such improvements is low-bandwidth approach. The latest attack method using such approach is low bit-rate type which exploits vulnerabilities of the application layer protocols to accomplish DoS attacks [1]. In this paper, we focus on the technique called Slow Read DoS Attack that has been designed by Sergey Shekyan [2]. This attack is that an attacker basically sends a legitimate HTTP request to target Web server and then very slowly reads the response. If an attacker sends many legitimate requests in this strategy, the Web server quickly reaches its maximum capacity and becomes unavailable for new connections by other legitimate clients. Moreover, it is very hard to detect if countermeasures do not monitor the network layer, because those requests are indistinguishable from other legitimate clients [3].

In Japan, some actual attacks using Slow Read DoS attack to the real on-line systems are reported but there are no open documents because they have clients' sensitive issues. In many cases, the targets are stock trade sites and on-line banking systems and damages of chance loss have caused [4]. In addition, there is a speculation for which Slow Read DoS Attack was used by Anonymous [5] to attack the site of JASRAC (Japanese Society for Rights of Authors, Composers and Publishers) for a protest against copyright protection in September, 2012 [6]. Thus the strategy of Slow Read DoS Attack evolves into the attack techniques with the actual damage.

In this paper, we analyze the effectiveness of Slow Read DoS Attack using the virtual network environment. We adopt *slowhttptest* as a general Slow Read DoS Attack scenario. It is freeware and available at [7]. We set the target Web server

using *Apache* which is most popular one [8], [9]. From our analysis, we found that there is the limitation of effectiveness of attack by a single attacker, and it is determined by the setup of Timeout parameter in Web server. And we also discovered the improvement attack technique using collusion attack scenario. As the result, we propose a new attack technique "Slow Read DDoS Attack (Slow Read Distributed DoS Attack)." Furthermore, we analyze the effectiveness of Slow Read DoS Attack to the Web server with *ModSecurity* which limits huge number of connections from the same source IP address [10]. As the result, we confirmed that Slow Read DoS Attack can be prevented by it completely. However, when we use our new attack technique, we can ignore the security setting of Web server with ModSecurity and we can succeed in attacking. We conclude that although there is a function which such countermeasure makes time length of attack success status restrict, an attack cannot be prevented fundamentally. In finally, we discussed the problems to solve for improvement of attack strategies and countermeasures against our proposed attack method.

2 SLOW READ DOS ATTACK

2.1 Outline of Slow Read DoS Attack

The Slow Read DoS Attack is one of Slow HTTP attacks. Slow HTTP attacks do not aim at the network layer like DoS and DDoS attacks, but exploit the application layer. If a HTTP request is not complete, or if the transfer rate is very low, the Web server keeps its resources busy waiting for the rest of data. Slow HTTP attacks are based on this fact. Figure 1 shows the outline of Slow Read DoS attack. Thus when the Web server keeps too many resources busy, this situation becomes like DoS attacks. To realize this malicious condition, the attacker can take following two types of techniques.

1) The technique of sending request slowly
2) The technique of reading response slowly

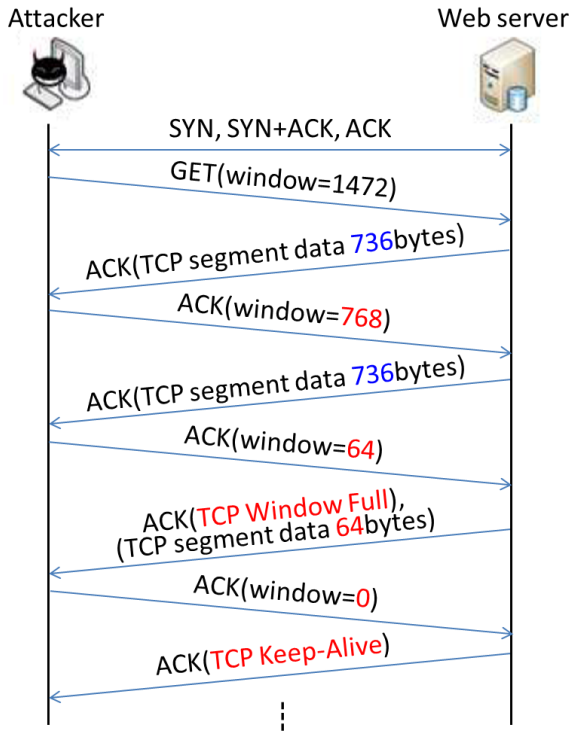

Figure 1. Outline of Slow Read DoS attack (packet flow)

Type 1) is well-known technique and Slowloris (also known as Slow Headers or Slow HTTP GET) or Slow HTTP POST attacks are famous. The Slow Read DoS Attack is categorized into type 2) and this is the latest technique. In this paper, we focus on type 2) strategy and technique of Slow Read DoS attack.

2.2 Basic Strategy

An attacker can deliver the Slow Read DoS attack by exploiting the flow control of TCP. Figure 1 also shows an example packet flow between an attacker and a target Web server in Slow Read DoS Attack procedure which can be captured by the network monitoring tool such as *Wireshark* [11]. First, the attacker sends a legitimate request after 3-way-handshake. After that, the attacker advertises the window size smaller than usual to make the HTTP response operation slow down. If the attacker advertises window size with zero, the Web server will stop sending data with keeping the connection. As the result, the attacker succeeds in Web server making resource waste.

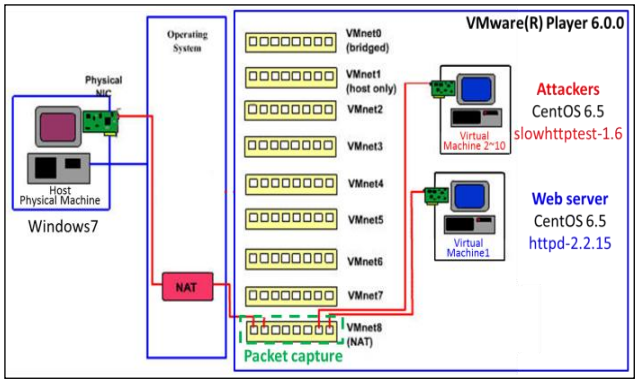

Figure 2. Experiment Environment [12]

Table 1. Directives of httpd.conf

Directive	Value
Timeout	60 (sec)
KeepAlive	Off

Table 2. Prefork MPM

Directive	Value
StartServers	8
MinSpareServers	5
MaxSpareServers	20
ServerLimit	256
MaxClients	256
MaxRequestPerChild	4000

Table 3. Parameters of slowhttptest

Parameter	Value
Number of attack connections	500
Receive window range	8-16 (byte)
Pipeline factor	1
Read rate from receive buffer	5 (byte/sec)
Connections Rate	50 (connections/sec)
Timeout for probe connection	10 (sec)
Using proxy	no proxy

3 PRELIMINARY

3.1 Experiment Environment

Figure 2 shows our experiment environment. We set 100 KB of Web page and use the Wireshark to observe packets between the attacker and Web server. Table 1 shows the default directives of "httpd.conf" which controls client connections to the Web server which is the attack target. Table 2 shows the default configuration of "prefork MPM (Multi-Processing Module)" which controls the generation of child processes when connections are established. Table 3 shows the parameters of slowhttptest which is a test tool of Slow Read DoS Attack.

Table 4. Parameters of Slow Read DoS Attack for each Experiment

Number of Experiment	Timeout	MC/SL
1	100 (sec)	200
2	200 (sec)	200
3	100 (sec)	300
4	200 (sec)	300
5	100 (sec)	600
6	200 (sec)	600
7	10 (sec)	200
8	10 (sec)	300

3.2 Max Client and Timeout

When Slow Read DoS Attack is delivered, service unavailable state of Web server is detected when the number of attack connections is reached the limit of Max Clients (MC in the following) in Table 2. When the total number of attack connection is larger than MC, there are no child processes to use. As the result, the Web server becomes service unavailable state. However, if the time of Timeout (Table 1) passes, the attack connection is disconnected compulsorily, and it will return to service available state. Thus, the Timeout parameter has a function which controls connection between them. And the value of MC controls the number of connections between the clients and the Web server.

3.3 Settings of Experiments

We fix the parameters of attacker as shown in Table 3 and set 8 types of target Web server as shown in Table 4. Note that parameter MC/SL means the value of MC and SL (Server Limit). We set them equal value because the setting of SL is not larger than the value of MC.

3.4 Definition of Attack Success

When conducting the Slow Read DoS Attack ex-

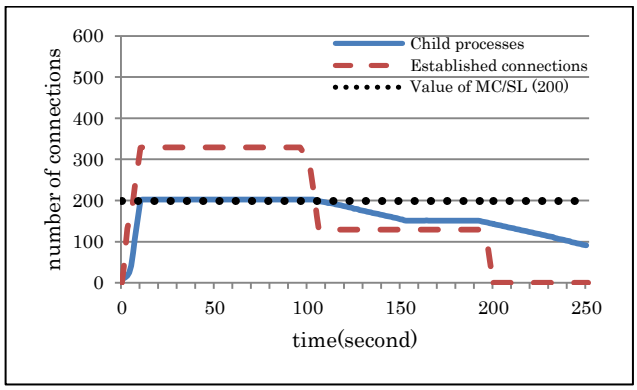

Figure 3. Experiment 1(Timeout 100, MC/SL 200)

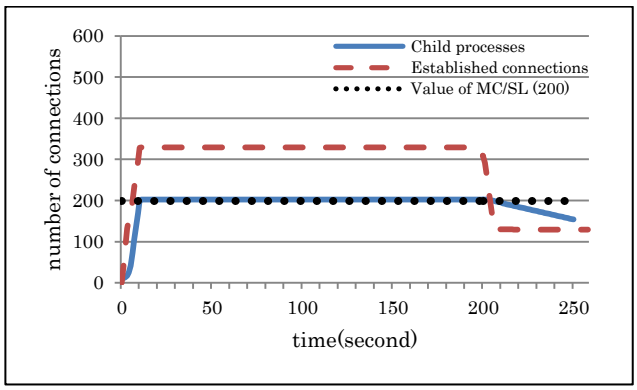

Figure 4. Experiment 2(Timeout 200, MC/SL 200)

periments, we define status of attack as follows.

- Attack success: Unacceptable new legitimate connections.
- Attack failure: Acceptable new legitimate connections.

"Attack success" means that the number of generated child processes is larger than or equal to the value of MC/SL. On the other hand, "Attack failure" means that the number of generated child processes is less than it. The effectiveness of attack success is estimated by time length (second) of maintaining service unavailable state.

4 EXPERIMENTS AND RESULTS

In the followings, we show the results of experiments as the graphs which show time transaction of the total number of child processes of Web server (blue line), and the total number of established connections of TCP which is connected to port #80 (red dashed line) after starting an attack at 0 second. And black dotted line shows the number of MC/SL. When the blue line is over the black dotted line, it is attack success.

4.1 Experiment 1 and 2

Figure 3 shows the result of Experiment 1. The total number of child processes reaches 200 which is the maximum value of MC/SL after 11 seconds. And we can succeed in the attack. However, by the setup of Timeout, the attack connections begin to be disconnected after 103 seconds, and it returned to the service available state. Therefore, the attacker is able to maintain the attack success status for 92 seconds. Though MC/SL is set to 200, the total number of established connections of TCP reaches 328. The reason for this situation is that these 128 (=328-200) of attack connections are contained by 3-way-handshake of TCP and treated as pending connections. By the setup of Timeout, the total number of established connections of TCP begins to decrease after 97 seconds. On the same time, pending connections are processed, but also disconnected after 193 seconds. Although the attack itself is ended after 10 (=500 (Total number of Attack connections) / 50 (Connections Rate)) seconds, the attack connections are processed until 193 seconds. As the result of Experiment 1, we can conclude that the attack succeeded between 11 and 103 seconds.

Figure 4 shows the result of Experiment 2. We set the value of Timeout to double (200 seconds) compared with Experiment 1. We can find that the attack success status is between 11 and 203 seconds, and the length is almost double of Experiment 1.

4.2 Experiment 3 and 4

We set the value of Timeout of Experiment 3 and 4 equal to Experiment 1 and 2 respectively, but are increased MC/SL to 300. Figure 5 and 6 shows the result of Experiment 3 and 4 respectively. From these results, they have the same tendencies as Experiment 1 and 2 but we can find that the number of child processes and the number of established connections of TCP are increased by the value of MC/SL. In Experiment 1 ~ 4, we used

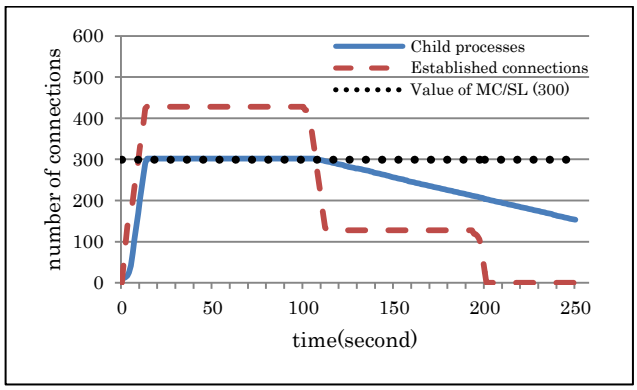

Figure 5. Experiment 3(Timeout 100, MC/SL 300)

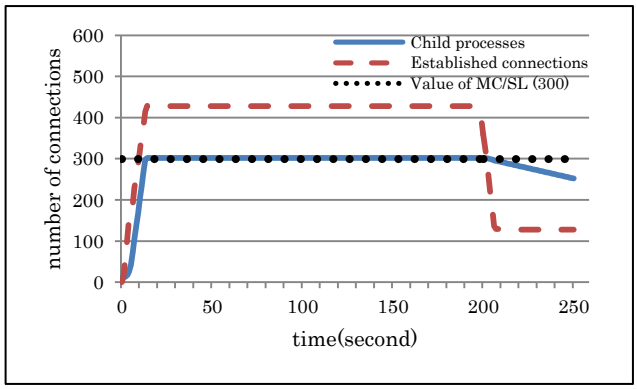

Figure 6. Experiment 4(Timeout 200, MC/SL 300)

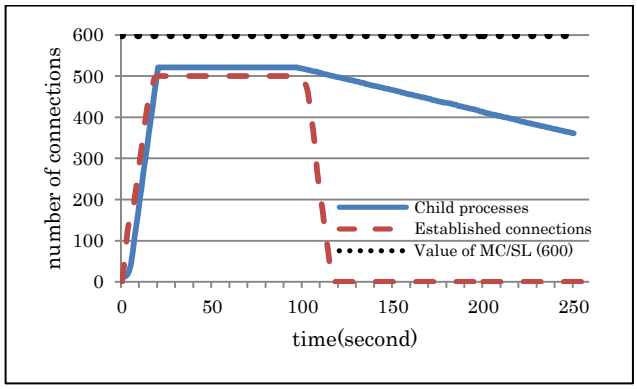

Figure 7. Experiment 5(Timeout 100, MC/SL 600)

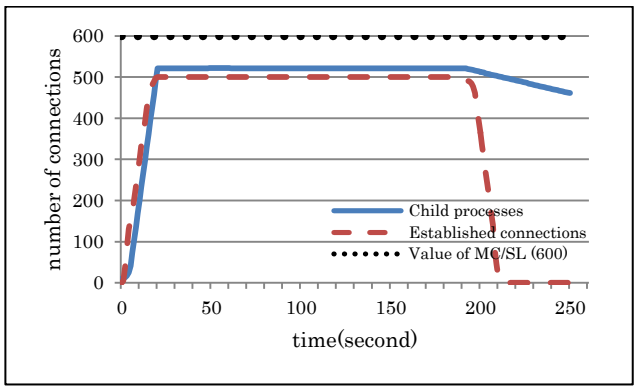

Figure 8. Experiment 6(Timeout 200, MC/SL 600)

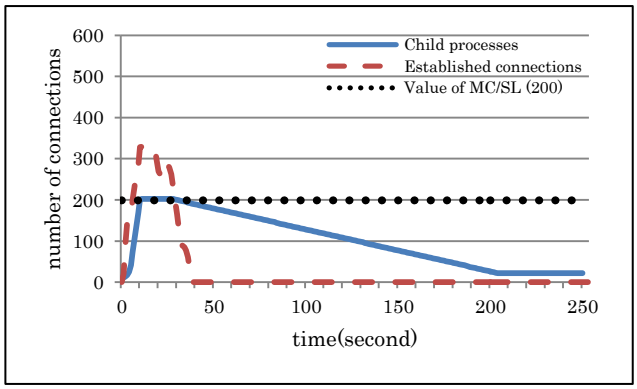

Figure 9. Experiment 7(Timeout 10, MC/SL 200)

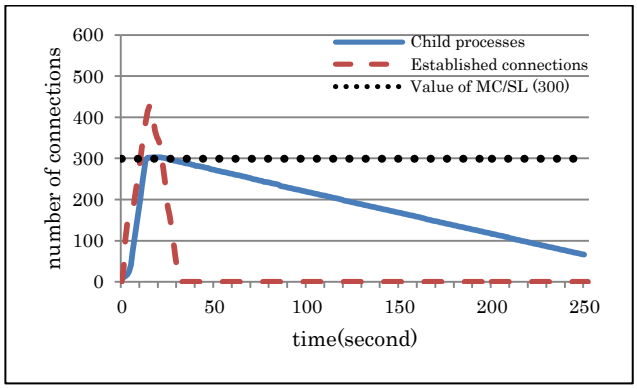

Figure 10. Experiment 8(Timeout 10, MC/SL 300)

500 of attack connections which are enough larger than MC/SL (300). Therefore, we can conclude that under such condition, the time length of attack success status is decided by the value of Timeout.

4.3 Experiment 5 and 6

We set the value of Timeout of Experiment 5 and 6 equal to Experiment 1 and 2 respectively, but are increased MC/SL to 600. Figure 7 and 8 show the result of Experiment 5 and 6 respectively. Naturally, we cannot succeed in the attack, because The Web server has enough resource against the total number of attack connections.

We can find two characteristics from these results. One is that the setup of Timeout also works after 98 seconds. Another is that we can find 520 of child processes, though the total number of attack connections is 500. This is because the number of MaxSpareServers is added as shown in Table 2.

4.4 Experiment 7 and 8

We set the value of MC/SL of Experiment 7 and 8 equal to Experiment 1 and 3 respectively, but are decreased Timeout to 10. Figure 9 and 10 show the result of Experiment 7 and 8 respectively. We can succeed in the attack. However, the attacker is able to maintain the attack success status for only 17 seconds in Experiment 7, and 8 seconds in Experiment 8, because the value of Timeout is extremely short. We can find that the value of Timeout is the main factor which determines the time length of attack success status.

4.5 Consideration

In order to analyze the relationship between the attack success status and the Web server parameters; Timeout and MC/SL, we conducted the attack simulations under the condition shown in Table 3 and 4. As the result, we can find following three factors that have determined attack success status.

1. The value of Timeout of Web server.
2. The value of MC/SL of Web server.
3. The total number of attack connections.

In general, the setup of Timeout is recommended for 10 seconds or more [13]. If Timeout is set short, Slow Read DoS attack can be prevented, but QoS (Quality of Service) is also reduced remarkably. If MC/SL is set large, Slow Read DoS Attack can be prevented, but there are limitations of resource and specification of the Web server. The number of attack connections is more effective factor, if it can be set huge comparing with Timeout and MC/SL. However, it depends on the attacker's cost. In addition, the attack connections from the same source IP address are easy to be detected. From these three reasons, we can conclude that the effectiveness of a single attacker's Slow Read DoS Attack is restrictive.

5 SLOW READ DDOS ATTACK

5.1 Outline of Proposal Technique

Since the attack connection is compulsorily disconnected by passing the Timeout, the attack success status of Web server returns to service available state. From the result of Experiment 8, we found that the countermeasure with 10 seconds of Timeout is effective against Slow Read DoS Attack which is parameterized as Table 3. Moreover, if the total number of attack connections is less than MC/SL, the attack cannot succeed. So, the effectiveness of attack by a single attacker is small as described in section 4.5. However, if another attacker sends new attack connections before previous attack connections are disconnected, it will be expected that the time length of attack success status can be maintained efficiently longer. Thus, we consider the scenario with which two or more attackers collude. In this paper, we call this attack technique "Slow Read DDoS Attack (Slow Read Distributed DoS Attack)."

Table 5. Variables

Variable	Explanation
t_0	Value of Timeout
t_z	Finish time of sending attack connections ($t_z = N / C$)
N	Total number of attack connections
C	Number of attack connections which send per second
K	Number of disconnected connections per second after t_0
M	Total number of connections that Web server can process (MC/SL)
$A_{(t)}$	Total number of attack connections which the attacker sent at time

5.2 Conditions for Attack Success Status

From the results of experiments shown in section 4, we can deduce the conditions of successful attack. Table 5 shows the variables. Let A be the total number of attack connections which connected to Web server. Then the condition of $M \leq A$ is necessary condition for successful attack, where M denotes the value of MC/SL. There are two cases of calculations for A under the conditions of t_0 and t_z. In the case of $t_0 \geq t_z$,

$$A(t) = C \times t \quad (1)$$

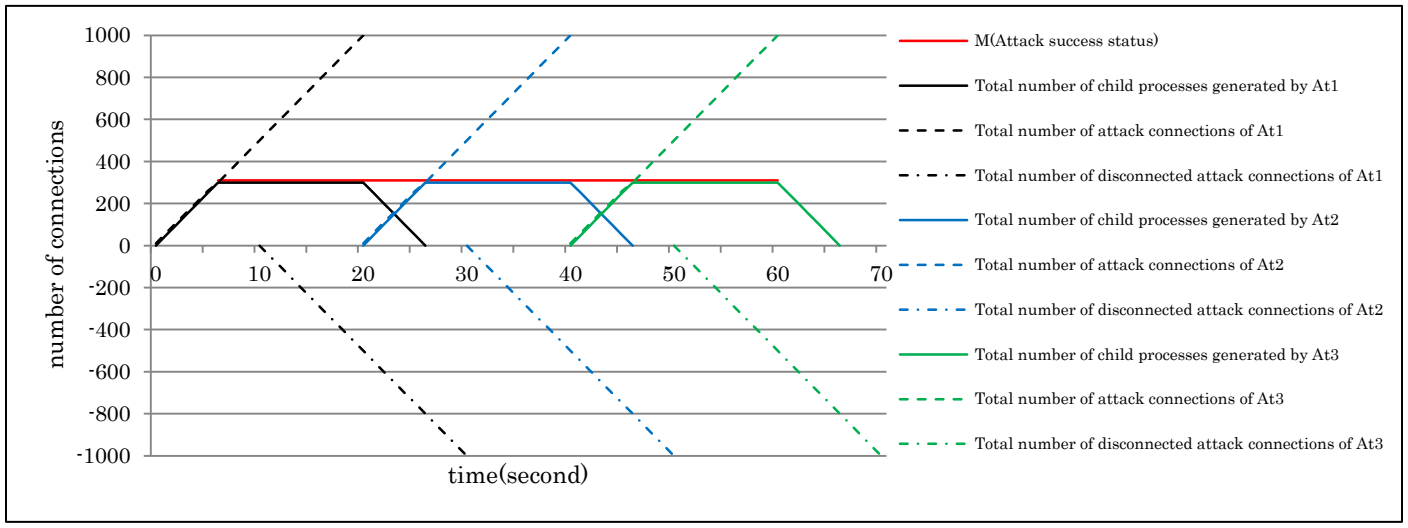

Figure 10. Attack Diagram

where C denotes the number of attack connection per second. In the case of $t_0 < t_z$,

$$A(t) = \begin{cases} C \times t & (t < t_0) \\ C \times t - K \times (t - t_0) & (t \geq t_0) \end{cases} \quad (2)$$

where t $(t \geq t_0)$ denotes time progress after attack starts $(t = 0)$. Eq. (1) is in the case that Timeout t_0 is enough large. Therefore, even if the attack finished at t_z by a single attacker, it will maintain the attack success status until t_0. However, since Timeout t_0 is shorter than t_z, in the case of Eq. (2), it is difficult to maintain the attack success status by a single attacker (see Experiment 8). Previous attack connections will be disconnected after Timeout t_0. This situation causes the limitation of attack effectiveness. In order to solve this problem, we use colluded attack scenario with some attackers.

The basic idea for maintaining attack success status is that following attacker begins to send new attack connections before former attacker's t_z. Therefore the colluded attackers can maintain attack success status by repeating it. Let us consider N attackers $At_1, At_2, \ldots, At_N$. Attacker At_n $(2 \leq n \leq N)$ begins his attack at ta_n (sec), which is calculated as follows.

Table 6. Parameters of Attack Simulation

Parameter		Value
Apache Web server	Timeout	10 (sec)
	ServerLimit	300
	MaxClients	300
Attackers	Connections Rate	50
	Number of attack connections	1,000

$$ta_n = \sum_{i=1}^{n-1} \frac{N_i}{C_i} \qquad (n \geq 2) \qquad (3)$$

where C_i and N_i denote the values of C and N which attacker At_i set respectively. Note that $ta_1 = 0$ (sec). For example, Figure 10 shows a theoretical attack diagram of the Slow Read DDoS Attack by three attackers deriving from Eq. (3).

We assumed that the target Web server is the same as Experiment 8, because it has the most resistant against Slow Read DoS Attack. The almost setting of attacker is the same as Section 4. However we set to $N = 1{,}000$, in order to hold the condition $t_0 < t_z$. As the results, we set the value of parameters to $C_1 = C_2 = C_3 = 50$, $N_1 = N_2 = N_3 = 1{,}000$, $t_0 = 10$, $t_z = 20$ and $M = 300$. Thus, we can deduce the attack diagram (Figure 10) of Slow Read DDoS Attack using these parameters. And from these settings, we can exp-

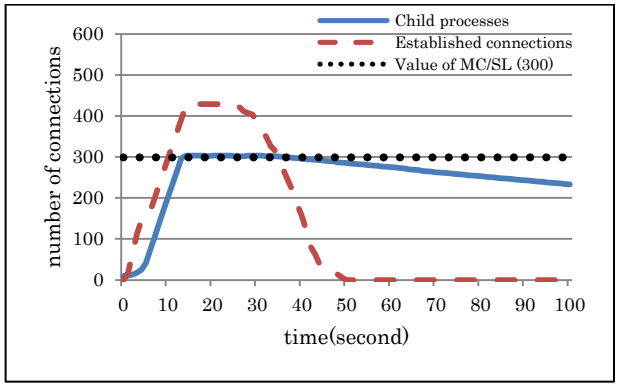

Figure 11. Simulation 1 (Attack result by At_1)

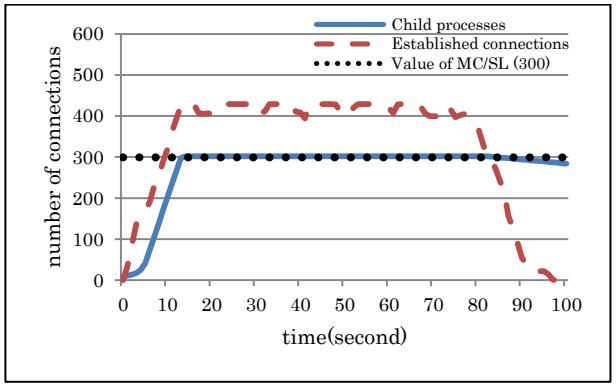

Figure 12. Simulation 2 (Attack result by At_1~At_3)

ect that attack success status can be maintained from 6 to 60 seconds. The parameters are summarized in Table 6.

6 ATTACK SIMULATIONS OF SLOW READ DDOS ATTACK

6.1 Outline of Attack Simulations

We set the parameters of attackers and Web server as Table 6. Before the Slow Read DDoS Attack simulation, in order to analyze the effectiveness using huge number of attack connections described in section 5.2, we conduct an attack simulation by a single attacker with $N = 1{,}000$ (Simulation 1). Next, we simulate the Slow Read DDoS Attack by three attackers following the attack diagram shown in Figure 10 (Simulation 2). From the attack diagram, each attacker's attack start time is set to $ta_1 = 0$ (sec), $ta_2 = 20$ (sec) and $ta_3 = 40$ (sec) using Eq. (3).

6.2 Results of Attack Simulations

Figure 11 shows the result of Simulation 1. We can find longer attack success status (20 seconds) than the result of Experiment 8 (8 seconds). We can confirm that increasing N is adequate for improving attack effect.

Figure 12 shows the result of Simulation 2. We can find that attack success status was maintained for 68 seconds (14~82 seconds). It is longer than the theoretical attack diagram shown in Figure 10 (54 seconds). From this result, we can conclude that the countermeasure with short Timeout do not work against our new attack technique.

6.3 Consideration

From the result of Simulation 1, we can consider that pending connections were also increased and they were newly processed even if Timeout passes. Because the total number of attack connections (N) was increased to 1,000 from 500, attack success status was maintained longer than Experiment 8.

We define following "attack rate" to evaluate the effectiveness of attack.

$$\text{attack rate} = \frac{\text{time length of attack success status}}{\text{number of attackers}} \quad \text{(sec/attacker)} \quad (4)$$

In Simulation 1, attack rate is 20.0 (sec/attacker). On the other hand, attack rate of Simulation 2 is 22.7 (sec/attacker). As the result, we can find that the Slow Read DDoS Attack is more efficient and lower-cost attack than Simulation 1.

In Simulation 2, we conducted the Slow Read DDoS Attack by three attackers. As the result, attack success status was maintained longer than Simulation 1. Therefore, we can consider that attack success status can be maintained for longer time, if three attackers attack repeatedly. We can find two facts between the attack diagram (Figure 10) and the result of Simulation 2 (Figure 12).

1. Time lag of child processes generation

From the result of attack shown in Figure 12, we can find that the attack starts to success after 14 seconds. On the other hand, from Figure 10, we expected after 6 seconds. Therefore there is 8 seconds of time lag. The reason is that we did not consider the delay time of generation of child processes which Web server generates. By predicting this time lag in advance, we will be able to set the attack start time to success exactly.

2. Influence of pending connections for establishment.

Pending connections will be generated, when the total number of attack connections (N) are set more than MC/SL. And they will be newly processed to establishment of connect after the time decided Timeout parameter for the former attack connections is passes. For this reason, we can consider that the pending connections can extend the time length of attack success status. Thus, it was extended longer than the theoretical attack diagram (Figure 10).

Therefore, there are two new problems for the improvement of our proposal attack.

1. Analysis of the child process generation.
2. Analysis of the pending connection processing.

If we can solve these problems, we can realize more precise and lower-cost attack.

7 MODSECURITY AND ITS EFFECTIVENESS

7.1 ModSecurity

"ModSecurity" is one of WAF (Web Application Firewall) which supports Apache HTTP Server, IIS and NGINX. It supplies real-time web application monitoring, logging, and access control. In this research, we use OWASP (Open Web Application Security Project) ModSecurity CRS (Core Rule Set) to control ModSecurity by setting configurable rule sets. OWASP ModSecurity CRS is distributed by Trustwave's SpiderLabs. The CRS provides configurable security rules such as follows [14].

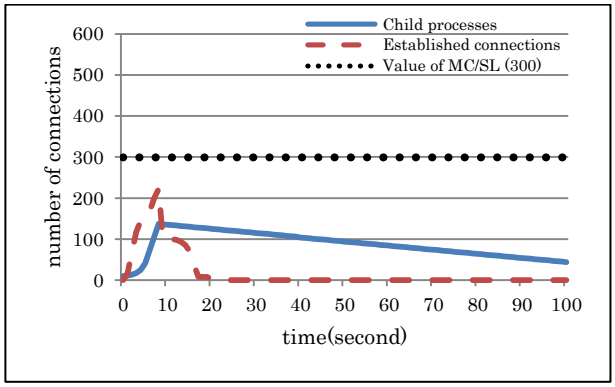

Figure 13. Experiment 9 (Timeout 10, MC/SL 300, Connection limit 100, Attack by At_1)

- HTTP Protection
- Real-time Blacklist Lookups
- HTTP Denial of Service Protections
- Common Web Attack Protection, etc.

In order to analyze the effectiveness of Web server with ModSecurity against the Slow Read DoS Attack, we focus on "HTTP Denial of Service Protections" to limit the number of connection from the same source IP address.

7.2 Outline of Experiment

We set the attacker's parameters as same as Table 3 in sections 3.1. And we set the target Web server same as Experiment 8; Timeout 10 and MC/SL 300. In addition, we use ModSecurity to limit the number of connections up to 100 from the same IP address. First, we conduct an experiment to analyze the effectiveness of Slow Read DoS Attack by a single attacker (Experiment 9). Next, we check the effectiveness of our proposal technique Slow Read DDoS Attack (Experiment 10) shown in section 5 with the same settings as Simulation 2.

7.3 Results of Experiments

Figure 13 shows the result of Experiment 9. We can see that ModSecurity functions after 8 seconds and the number of established connections are dramatically decreased. We can confirm that after ModSecurity starts, it can react immediately to the increase in attack connections. But since there is time lag of its starting, the Web server allows many generations of child processes which are

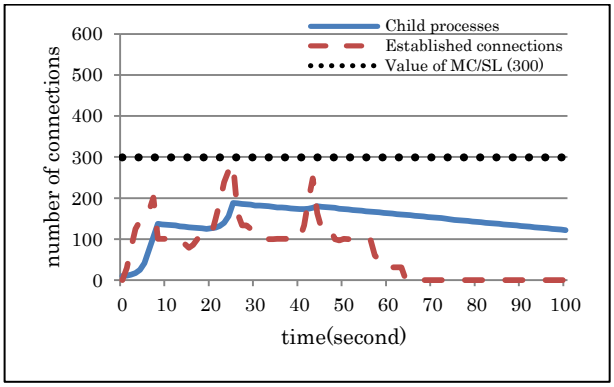

Figure 14. Experiment 10 (Timeout 10, MC/SL 300, Connection limit 100, Attack by $At_1 \sim At_3$)

more than the limitation number. As the result, 138 child processes are generated. Since the reduction in child process follows the setup of Timeout, in the case of Experiment 9, child processes decrease at rate 1 (process/sec). In the settings of Experiment 9, we cannot succeed in the attack at all from above reasons. Therefore, we can conclude that ModSecurity has enough effectiveness against a simple Slow Read DoS Attack scenario.

Figure 14 shows the result of Experiment 10. We can see that the attack also did not succeed at all with same reasons described above. From the time transaction of total number of established connections, we can see that ModSecurity has enough effectiveness in the same way of Experiment 9. In addition, by the time lag of ModSecurity starting, the total number of child processes increased more than 100 at 8 seconds and 25 seconds. They are in a situation as the purpose of the Slow Read DDoS Attack. However, in the sense of increasing the number of child process, contribution of At_3 is smaller than At_2. This is because it is set off against reduction in child processes generated by At_1 and At_2. Therefore we can expect that the attack can be succeeded if the interval of each attacker's start time (ta_n) is closer than estimation using Eq. (3).

To confirm our assumption, we did two types of experiments with heuristic attack setup. They are Condition 1 with ta_1 = 0 (sec), ta_2= 10 (sec), ta_3= 20 (sec), and Condition 2 with ta_1 = 0 (sec), ta_2= 5 (sec), ta_3= 10 (sec).

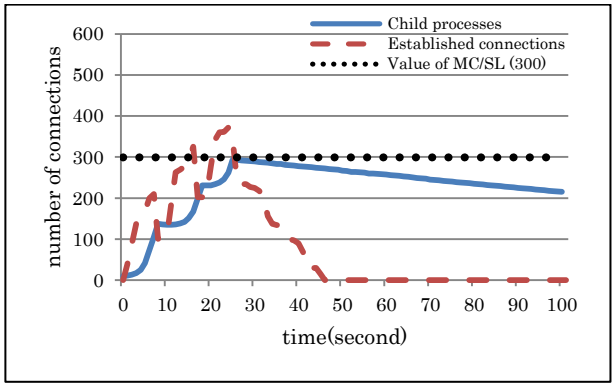

Figure 15. Condition 1 ($ta_1 = 0, ta_2 = 10, ta_3 = 20$)

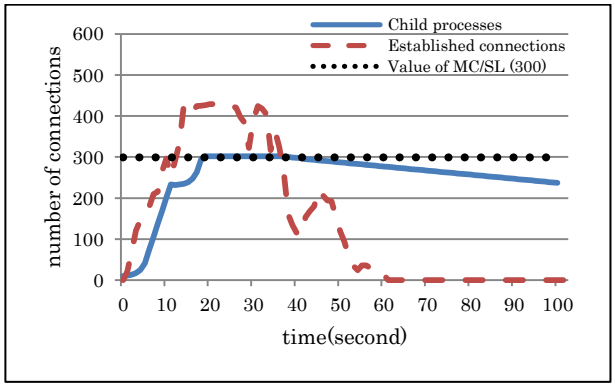

Figure 16. Condition 2 ($ta_1 = 0, ta_2 = 5, ta_3 = 10$)

Figure 15 shows the result of Condition 1. From this result, we can confirm that the total number of child processes did the monotone increase without influence of reduction of Timeout and disconnecting of attack connections by ModSecurity. As the result, the total number of child processes reached 293 at 25 seconds. Unfortunately, it has not yet resulted in the attack success.

Figure 16 shows the result of Condition 2. We can find that attack success status is maintained for 17 seconds (19~36 seconds) and attack rate was 5.7 (sec/attacker). Therefore it shows that our assumption was adequate. However, the attack diagram was derived heuristically from our attack experiments. Development of theoretical derivation of attack diagram against Web server with ModSecurity is our future work. We can see that the total number of established connections is increased at 31 seconds and 45 seconds in Figure 16. We can consider that this is because the pending connections which were waiting for

establishment were newly processed as shown in the subject 2 of section 6.3.

As the results, we can conclude that the security setting of Web server which applying the short value of Timeout with ModSecurity has enough effectiveness against the Slow Read DoS Attack and simple Slow Read DDoS Attack. However, from the results of Condition 1 and 2, we can expect that some techniques can lose effectiveness of these countermeasures. The heuristic type of the improvement technique is already shown above. The algorithmic type which uses many colluded attack groups is shown in section 8.

8 IMPROVEMENT OF SLOW READ DDOS ATTACK

8.1 Attack Strategy against ModSecurity

In this section, we consider how to deceive the function of ModSecirity. In a simple way, we take a technique reducing the number of connections from the same IP address by increasing the number of attackers. The advantage of this technique is able to predict the effectiveness of attack easily by applying the technique shown in section 5.2 to attackers who are member of the same group. This is more positive than heuristic type as shown in previous section. On the other hand, if the setup of limitation number in ModSecurity CRS is unknown, the number of attackers is not decided. So, it is necessary to perform the attacks such as Experiment 9 in previous and predict the number of limitation. And it is easy to collect many attackers whose IP address is unique, if we use botnet which is popular in DDoS attack [15]. So, it is easily considered from the above discussions to satisfy the condition for the effective improved Slow Read DDoS Attack.

In the followings, we assume that the settings of Web server are known to attacker and it is the same as Experiment 9 in section 7.2. Since the value of limitation number in ModSecurity is 100, the maximum number of attack connections which one attacker can generate is 100. And since MC/SL = 300, one group needs to consist of three attackers (300/100=3) at least. Therefore, the following composition is the minimum attack unit.

- Attack Group 1 (Atg_1): ($\mathrm{At}_{11}, \mathrm{At}_{12}, \mathrm{At}_{13}$)
- Attack Group 2 (Atg_2): ($\mathrm{At}_{21}, \mathrm{At}_{22}, \mathrm{At}_{23}$)
- Attack Group 3 (Atg_3): ($\mathrm{At}_{31}, \mathrm{At}_{32}, \mathrm{At}_{33}$)

Total: 9 attackers.

Thus, minimum attack unit can be easily constituted using the information of limitation number of ModSecurity and the value of MC/SL. When many attackers can be prepared rather than minimum, it is obvious that more efficient attack can be performed.

Each attack group attacks according to the attack diagram which is shown in figure 10. And the attackers of each group conduct simultaneously. In other words, in order to ignore the function of ModSecurity, we increase the number of attackers to attack at the same time. And in order to maintain the attack success status for a long time, we have to increase other attack groups whose IP address are different previous attackers.

8.2 Outline of Attack Simulations

We set the parameters of Web server and attacker as Experiment 9. We assumed that three attack groups and one group consist of three attackers as shown in section 8.1. In the following attack simulations, our purpose is to analyze the total number of attack connection $\mathrm{N_i}$ of each attacker for successful attack. Therefore each group's attack start time follows the attack diagram shown in Figure 10. So, $\mathrm{tg}_1 = 0$ (sec), $\mathrm{tg}_2 = 20$ (sec) and $\mathrm{tg}_3 = 40$ (sec), where $\mathrm{tg_n}$ denotes attack start time of attack group n. We conducted two types of attack simulations; Simulation 3 and Simulation 4.

Simulation 3 is for the minimum attack unit. From Simulation 2 (see Figure 12), we obtained the successful attack result near theoretical estimation under the condition of $\mathrm{N_i} = 1{,}000$. Since 1,000 of attack connections from one group are necessary, 340 of attack connections per attacker are assigned. So, in Simulation 3, we set the attack condition

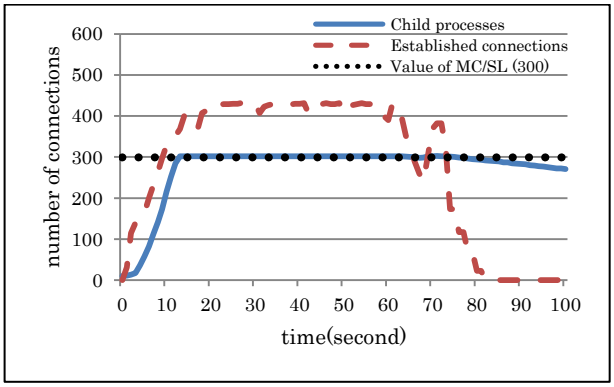

Figure 17. Simulation 3 (N_{11}~ N_{33} = 340)

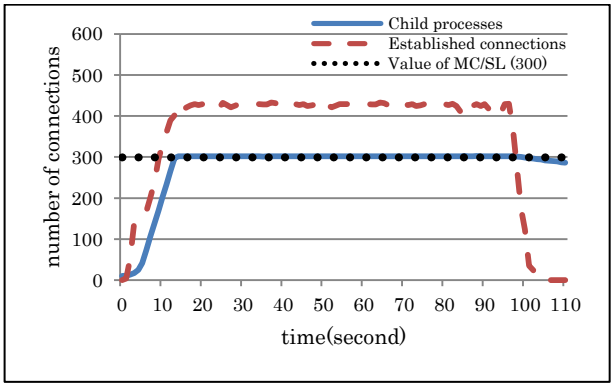

Figure 18. Simulation 4 (N_{11}~ N_{33} = 1,000)

with N_{11}~ N_{33} = 340 to expect the effectiveness such as Condition 1 shown in section 7.3 (Figure 15).

Simulation 4 denotes a generous attack condition. The result of Simulation 2 shows that the condition of N_i = 1,000 is not high cost for each attacker. The purpose of this simulation is to analyze the effectiveness of generation of pending connections, and its influence for attack result. So, in Simulation 4, we set the attack condition with N_{11}~ N_{33} = 1,000.

8.3 Results of Simulations

Figure 17 shows the result of Simulation 3. We can find that the attack success status was maintained for total 56 seconds (13~65 seconds and 69~73 seconds). From 66 seconds, the number of child processes became lesser than 300 because of Timeout and ModSecurity, and changed to attack failure status. After 69 seconds, it returned to attack success status because of the effectiveness of Atg_3 's established connections. However, as the same situation as Condition 1 shown in Figure 15, the contribution of Atg_3 is lesser than Atg_2. So, it maintained attack success status for only four seconds. As the result, we can judge that Simulation 3 succeeded in the attack.

Figure18 shows the result of Simulation 4. On the whole, this simulation succeeded in the attack for 83 seconds. If we compare with Simulation 2, assuming one group to be one attacker, this result means that Simulation 4 used three times as many attack connections in Simulation 2. From the view point of this way, the condition of Simulation 4 is not efficient than Simulation 2, although there is a countermeasure of ModSecurity. Therefore it is thought that there is a method of determining more effective N and this is our future work.

As the results, we can confirm that the countermeasure which limits the number of connections from the same source IP address can be ignored by our improved new attack technique.

8.4 Consideration

From the results of Simulation 3 and 4, the attack rate of Simulation 3 is 6.2 (sec/attacker), and of Simulation 4 is 9.2 (sec/attacker). Comparing with Simulation 2, the attack cost is rose by applying ModSecurity on Web server. In this point, there is effectiveness as a countermeasure. However, ModSecurity cannot prevent the attack at all and "HTTP Denial of Service Protections" has not enough effectiveness against improved Slow Read DDoS Attack.

As shown in section 8.1, in order to hold attack success status longer than Simulation 4, we need to prepare other attack groups whose IP addresses are unique each other. On the other hand, if ModSecurity is not used, two attackers can maintain in the attack success status forever by attacking by turns in theoretically. If ModSecurity is used, even if we attack by improved Slow Read DDoS Attack with botnet, the time of attack success status is limited. This is the significant point of ModSecurity.

Also in Simulation 3 and 4, we can conclude that the pending connections are important factor. We already pointed out in section 6.3, analysis of the pending connections processing is useful for improvement of attack technique or defense strategy.

9 DISCUSSIONS

The important feature of Slow Read DoS Attack is that the target Web server is not down. From our attack simulations, it is clear from the fact that the target Web server certainly returned to service available state after the attack. Conversely, it is difficult for administrators to detect the attack. Especially, since legitimate requests are sent, it is expected that the signature type IDS is impossible to detect the attack.

In our attack simulations, in order to attack strategy simply, the following conditions were given.

1. The value of C (the number of attack connections which send per second) is fixed to 50.
2. The window size is fixed to zero.

When the target Web server is Apache, the maximum number of generated child processes per second is 32 [16]. Thus, under the condition of C=50, 18 of attack connections are processed per second as pending connections. Since child process is valid within the period decided by Timeout, if we can control the rate of generating pending connections by C, we could develop more effective attack method controlling the pending connections.

In our attack scenario, the attacker advertises "window size = 0." However, this is not necessary action in actual Slow Read DoS Attack. In fact, it is easy to be detected as malicious action. So, the value of window size should be set to a minimum necessary. As a result, although the effectiveness of attack is inferior to our simulations, the risk of attack detection will become small. In addition, the technique to which the value of window size is changed flexibly in a session is also considered. The evaluation to "Adaptive Slow Read DoS Attack" which changes the value of C and window size is our future work.

We also showed that improved Slow Read DDoS Attack is effective even if the target Web server uses secure modules. This attack requires the systematized attacker group. However, in the latest cyber-attack and cyber-crime, the executions by the systematized group are general cases. We should recognize that this attack scenario is a real threat.

As already we described above, the construction of IDS against Slow Read DoS and Slow Read DDoS Attack is very difficult. However, it is thought that the behaviour of adaptive attack shown above has some characteristics. So, an anomaly type IDS may be able to be constructed by discovering such features. In other way, TCP acceleration will prevent Slow Read DoS Attack from foundation. TCP acceleration is technique to achieve better throughput between client and server using TCP tuning [17]. There is a method to decide the value of window size based on the real-time measurement of packet arrival time [18], and it can be an effective countermeasure against our proposal attack method. This is also our future work.

10 CONCLUSIONS

In this paper, we analyzed the effectiveness of Slow Read DoS Attack by virtual network environment. From results, we concluded that the attack by a single attacker is not so efficient. However, we can derive the improvement of Slow Read DoS Attack and develop Slow Read DDoS attack. And we derived the attack diagram which maximizes the effectiveness of Slow Read DDoS Attack. We confirmed it by computer simulations. In addition, we conducted the attack simulations against the Web server with secure module, ModSecurity. ModSecurity can limit the length of attack success status however, attack itself cannot be prevented. As the result, we succeeded in improving the attack technique whose effectiveness is same level in the case of attack against the target Web server without secure

modules. We summarized our discussions and future works in section 9. And we concluded that the analysis of generation of pending connections and the time lag of starting security modules or child process are important factor to improve the attack technique and to develop countermeasures.

Note that our attack simulations are done in only one-hop virtual environment in order to be easy to analyze (see Figure 2). Therefore, the attack will affect differ in actual internet environment. Specifically, we have to add followings as attack factors.

- Existence of intermediate servers and routers.
- Existence of legitimate users who has already connected.

Since the existence of intermediate servers and routers affects the communication speed, we should adjust the value of C (The number of attack connections which send per second) and N (The total number of attack connections) for successful attack. Depending on the number of already connected by general users, our attack may be successful with lesser cost. The analysis of more actually based threat is also our future work.

11 REFERENCES

1. Cambiaso Enrico, Papaleo Gianluca, Chiola Giovanni and Aiello Maurizio, "Slow DoS attacks: definition and categorisation," Int. J. Trust Management in Computing and Communications, Vol. 1, No. 3/4, pp. 300-319 (2013).
2. Sergey Shekyan, "Are you ready for slow reading?," QUALYS BLOG, Available: https://community.qualys.com/blogs/securitylabs/2012/01/05/slow-read (Last access: Dec. 12, 2014)
3. Kelly Jackson Higgins, "New Denial-Of-Service Attack Cripples Web Server By Reading Slowly," InformationsWeek DarkReading (Internet article), Available: http://www.darkreading.com/attacks-breaches/new-denial-of-service-attack-cripples-we/232301367 (Last access: Dec. 12, 2014)
4. Comments of Mizuho Research Institute for our presentation "Anlysis of Slow Read DoS Attack and countermeasure" at Japanese domestic conference, Computer Security Symposium (2014).
5. Anonymous Operation Japan, twitter account "@OP-japan."
6. Japanese bulletin board, 2ch.net.
7. Sergey Shekyan, "slowhttptest," Available: https://code.google.com/p/slowhttptest (Last access: Dec. 12, 2014)
8. W3Techs, "Most popular web servers," Available: http://w3techs.com (Last access: Dec. 12, 2014)
9. Apache, Available: http://httpd.apache.org (Last access: Dec. 12, 2014)
10. ModSecurity, Available: http://www.modsecurity.org (Last access: Dec. 12, 2014)
11. Wireshark, Available: https://www.wireshark.org (Last access: Dec. 12, 2014)
12. ExtremeTech, "Virtual Machines & VMware Part II," Available: http://www.extremetech.com/computing/72268-virtual-machines-vmware-part-ii (Last access: Dec. 12, 2014)
13. Noopy Zang, "Tuning of Apache (in Korean blog)," Available: http://openlife.tistory.com/340 (Last access: Dec. 12, 2014)
14. OWASP ModSecurity Core Rule Set Project, Available: https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project (Last access: Dec. 12, 2014)
15. Esraa Alomari, B. B. Gupta, Shankar Karuppayah, "Botnet-based Distributed Denail of Service(DDoS) Attacks on Web Servers," Classification and Art, Int. Journal of Computer Applications (0975-8887), Vol. 49, No.7, pp. 24-32 (2012).
16. Apache Performance Tuning, Available: http://httpd.apache.org/docs/trunk/en/misc/perf-tuning.html (Last access: Dec. 12, 2014)
17. Sameer Ladiwala, Ramaswamy Ramaswamy, Tilman Wolf, "Transparent TCP acceleration," Computer Communications 32, pp. 691-702 (2009).
18. Takashi Isobe, Naoki Tanida, Yugi Oishi and Ken-ichi Yoshida, "TCP Acceleration Technology for Cloud Computing: Algorithm, Performance, Evaluation in Real Network," Proceedings of the 2014 International Conference on Advanced Technologies for Communication, pp. 714-719 (2014).
19. Evan Damon, Julian Dale, Evaristo Laron, Jens Mache, Nathan Land and Richard Weiss, "Hands-On Denial of Service Lab Exercises Using Slowloris and RUDY," Proceedings of the 2012 Information Security Curriculum Development Conference, pp. 21-29 (2012).
20. Cambiaso Enrico, Papaleo Gianluca, Chiola Giovanni and Aiello Maurizio, "Taxonomy of Slow DoS Attacks to Web Applications," International Conference on Security in Computer Networks and Distributed Systems 2012, Communications in Computer and Information Science 335, pp. 195-204 (2012).
21. Hiroshi Kurakami, "The advanced DDoS attack and countermeasure (in Japanese)," IPSJ (Information Processing Society of Japan) Magazine, Vol.54, No.5, pp. 475-480 (2013).
22. Jonathan Lemon, "Resisting SYN flood DoS attacks with a SYN cache," Proceedings of the BSD Conference 2002 on BSD Conference (2002).

23. Junhan Park, Keisuke Iwai, Hidema Tanaka and Takakazu Kurokawa, "Study of Slow Read DoS Attack on Web server (in Japanese)," Proceeding of the 31st Symposium on Cryptography and Information Security, 2C1-4 (2014).
24. Junhan Park, Keisuke Iwai, Hidema Tanaka and Takakazu Kurokawa, "Analysis of Slow Read DoS Attack," 9th ACM Symposium on Information Computer and Communications Security, Poster Session (2014).
25. Junhan Park, Keisuke Iwai, Hidema Tanaka and Takakazu Kurokawa, "Analysis of Slow Read DoS Attack and Countermeasures (in Japanese)," Proceeding of the Computer Security Symposium, pp. 354-361 (2014).
26. Junhan Park, Keisuke Iwai, Hidema Tanaka and Takakazu Kurokawa, "Analysis of Slow Read DoS Attack," Proceeding of the International Symposium on Information Theory and Its Applications, pp. 60-64 (2014).
27. Junhan Park, Keisuke Iwai, Hidema Tanaka and Takakazu Kurokawa, "Analysis of Slow Read DoS Attack and Countermeasures," Proceeding of the International Conference on Cyber-Crime Investigation and Cyber Security, pp. 37-49 (2014).
28. Stephen Specht and Ruby Lee, "Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures," 2004 International Workshop on Security in Parallel and Distributed Systems, pp. 543-550 (2004).
29. Takeshi Yatagai, Takamasa Isohara and Iwao Sasase, "Dectection Technique of HTTP-GET Flood Attack Based on Analysis of Page Access Behavior (in Japanese)," IPSJ (Information Processing Society of Japan) SIG Technical Report, 2007-CSEC-37, pp. 33-38 (2007).
30. Ha.ckers, Slowloris HTTP DoS, Available: http://ha.ckers.org/slowloris (Last access: Dec. 12, 2014)
31. KISA (Korea Internet Security Agency), "The response guide against DDoS attack (in Korean)," Technical Report, KrCERT-TR-2012-002, Available: http://www.krcert.or.kr (Last access: Dec. 12, 2014)
32. Ronen Kenig, "Why Low & Slow DDoS Application Attacks are Difficult to Mitigate (blog)," Available: http://blog.radware.com/security/2013/06/why-low-slow-ddosattacks-are-difficult-to-mitigate (Last access: Dec. 12, 2014)
33. VMware Player, Available: http://www.vmware.com/jp/products/player (Last access: Dec. 12, 2014)