Analysis of Slow Read DoS Attack and Countermeasures

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 the DoS / DDoS Attack strategy become more effective and more complex. In our research, 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 legitimate requests, he can keep many open connections to Web server and eventually cause DoS situation. In this paper, we analyze the effectiveness of the Slow Read DoS Attack using the virtual environment. As a result, we found that the 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 the Slow Read DoS Attack technique, we can also find that these countermeasures are not effective against distributed Slow Read DoS Attack (Slow Read DDoS 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 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/DDoS attacks, and many solutions are proposed. One of such solution is low-bandwidth approach. The latest attack method using such approach is low bit-rate type which exploits vulnerabilities of 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 the Web server and then very slowly reads the response. If an attacker sends many legitimate requests, the Web server quickly reaches its maximum capacity and becomes unavailable for new connections by legitimate clients. Moreover, it is very hard to detect these attacks if we do not monitor the network layer, because those requests are indistinguishable from other legitimate clients [3].

In this paper, we analyze the effectiveness of the 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 [4]. We set the target Web server using *Apache* which is most popular one [5]. From our analysis, we found that there is the limitation of effectiveness of attack by a single attacker, and it is determined by the setting 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”.

Furthermore, we analyze the effectiveness of the Slow Read DoS Attack to the Web server with the *ModSecurity* which limits huge number of connections from the same source IP address [6]. As a result, we confirmed that the 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 the ModSecurity and we can succeed in attacking. Although there is a function which such countermeasure makes time length of attack success status restrict, an attack cannot be prevented fundamentally. We discussed the relation between the effectiveness of IDS (Intrusion Detection System) and sophisticated Slow Read DoS Attack. And we concluded that the analysis of generation of pending connections using the time lag of starting security modules or child process are important factors to improve of the attack technique and development of countermeasures.

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 / 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 the data. Slow HTTP attacks are based on this fact. 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

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.

2.2 Attack Strategy

The attacker can deliver the Slow Read DoS attack by exploiting the flow control of TCP. 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.

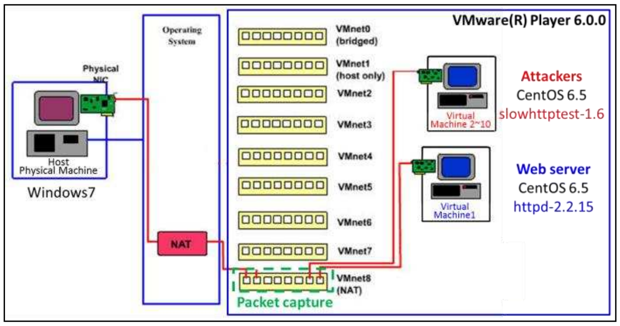

Figure 1. Experiment Environment

Table 1. Directives of httpd.conf

Directive	Value
Timeout	60
KeepAlive	Off

Table 2. Prefork MPM

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

If the attacker advertises window size with zero, the Web server will stop sending data with holding the connection. As a result, the attacker succeeds in Web server making resource waste.

3 ANALYSIS OF THE SLOW READ DOS ATTACK EFFECT

3.1 Preliminary

3.1.1 Experiment Environment

Figure 1 shows experiment environment. We set the Web page size 100KB and use *Wireshark* [8] to observe packet between the attacker and Web server. Table 1 shows the default directives of "httpd.conf" which controls a connection with a client in the configuration of the Web server which is the attack target. Table 2 shows the default configuration of "prefork

Table 3. Attack Options of the Slow Read DoS Attack

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

Table 4. Attack Options of the Slow Read DoS Attack

Number of Experiment	Timeout	MC/SL
1	100 sec	300
2	200 sec	300
3	100 sec	600
4	10 sec	300

MPM (Multi Processing Module)" which controls the generation of child processes when a connection is established. Table 3 shows the attack options of the Slow Read DoS Attack.

3.1.2 Function of Parameter

When the Slow Read DoS Attack is delivered, a service unavailable state of Web server is detected when the attack connection 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 a 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, 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.

3.1.3 Setting of Experiment

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

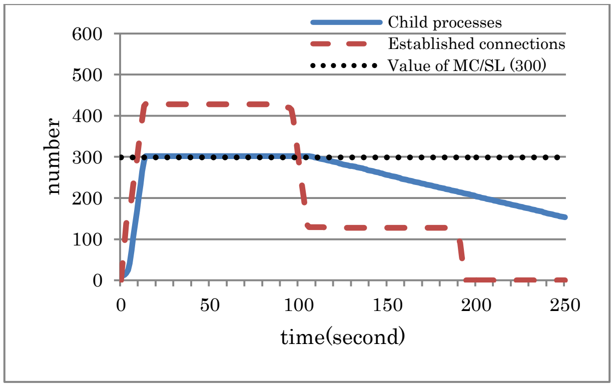

Figure 2. Experiment 1(Timeout 100, MC/SL 300)

3.1.4 Definition

When conducting the Slow Read DoS Attack experiment, we define the 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 the value of MC/SL. The effectiveness of successful attack is estimated by time length (second) of maintaining service unavailable state.

3.2 Results of Experiments

In the followings, the graph of an experimental result shows time transaction of the total number of child processes of Web server (blue line), and the total number of established connections of TCP to the port #80 (red dotted line) after starting an attack at time 0 second.

3.2.1 Experiment 1, Experiment 2

Figure 2 shows the result of the experiment 1. The total number of child processes reaches 300 which is the value of MC/SL after 14 seconds. As a result, we can succeed in attack.

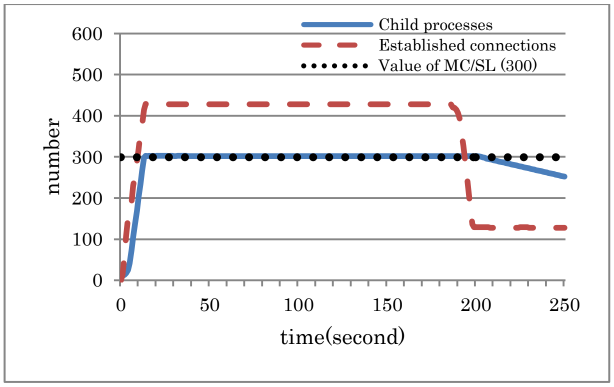

Figure 3. Experiment 2(Timeout 200, MC/SL 300)

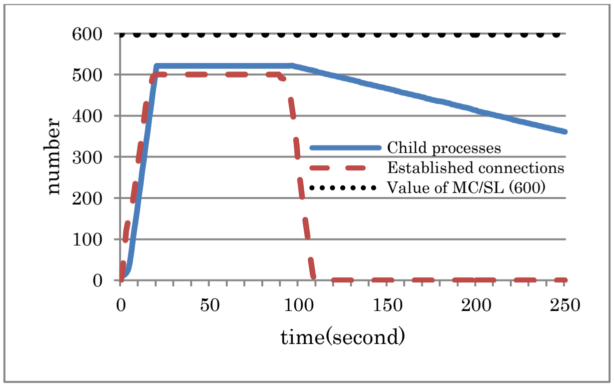

Figure 4. Experiment 3(Timeout 100, MC/SL 600)

However, by setup of Timeout, attack connections begin to be disconnected after 107 seconds, and it returned to the service available state. Therefore, the attacker is able to maintain attack success status for 93 seconds. Though MC/SL is set to 300, the total number of established connections of TCP becomes 428. The reason for this situation is that these 128 (=428-300) attack connections are contained by 3-way-handshake of TCP and treated as pending connections. By setup of Timeout, the total number of TCP established connections also decreases after 93 seconds. On the same time, pending connections are processed, but also disconnected after 187 seconds. Although the attack itself is ended after 10 (=500 (Total number of Attack connections) / 50 (Connections Rate)) seconds, the attack connections are effective until 194 seconds. As the result of experiment 1, we can conclude that the attack succeeded between 14 and 107 seconds.
Figure 3 shows the result of the experiment 2. We set the value of Timeout to double (200 seconds) compared with the experiment 1. We can find that the attack success status is between 14 and 201 seconds, and the length is almost double of experiment 1.

3.2.2 Experiment 3

Figure 4 shows the result of experiment 3. Naturally, we cannot succeed in attack, because the Web server has an enough resource against the total number of attack connections. We can find two characteristics from the result. 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 MaxSpareServer is added as shown in Table 2.

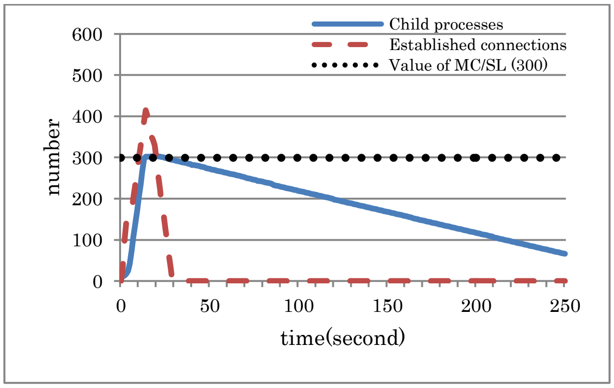

Figure 5. Experiment 4(Timeout 10, MC/SL 300)

3.2.3 Experiment 4

Figure 5 shows the result of experiment 4. We can succeed in attack. However, the attacker is able to maintain the attack success status for just 8 seconds, since the value of Timeout is extremely short. We can find that the value of Timeout is a main factor which determines the time length of attack success status.

3.3 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. As a 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 of attacker.

In general, the setup of Timeout is recommended for 10 seconds or more. If Timeout is set short, the 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 of Web server. The number of attack connections is more effective if it is set huge from the relationship between Timeout and MC/SL. However, it depends on the attacker's cost. In addition, the attack connections from the same IP address is easy to be detected. From these three factors, we can conclude that the effectiveness of a single attacker's Slow Read DoS Attack is restrictive.

4 PROPOSAL OF SLOW READ DDOS ATTACK

4.1 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 experiment 4, 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 success. So, the effectiveness of attack by a single attacker is small as described in section 3.3. However, if another attacker sends new attack connections before attack connections are disconnected, it will be expected that the 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”.

Table 5. Variables of Calculation

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

4.2 Conditions for Attack Success Status

From the result of experiments shown in section 3, 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. There are two cases of calculations for A under the conditions of t_0 and t_z. In the case of $t_0 \geq t_z$,

$$A(t) = C \times t \tag{1}$$

In the case of $t_0 < t_z$

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

where $(t \geq t_0)$ denotes time progress after attack starts $(t = 0)$.

4.3 Proposal Technique

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 4). Former attack connections will be disconnected

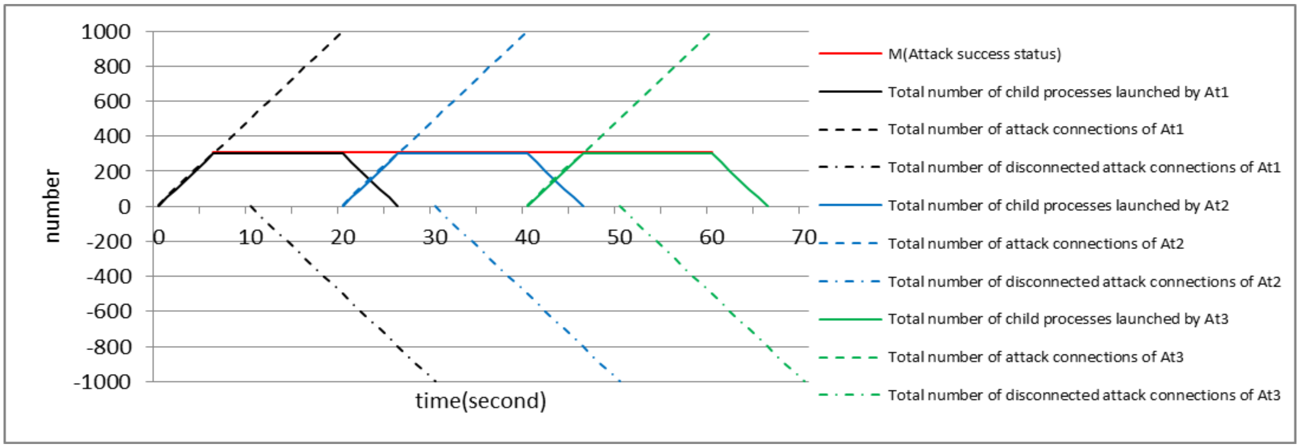

Figure 6. Attack Diagram

after Timeout t_0. This situation causes the limitation of effectiveness of attack. In order to solve this problem, we consider colluded attack scenario with some attackers.

The basic idea for maintain the attack success status is that following attacker begins to send new attack connections before former attacker's t_z. Therefore the collusion attackers can maintain the 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 the attack at ta_n, which calculated as follows.

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

where C_i and N_i denote the values of C and N which attacker At_i set respectively. Note that $ta_1 = 0$. For example, Figure 6 shows a theoretical attack diagram of the Slow Read DDoS Attack by three attackers. We assumed that the target Web server is the same as experiment 4, because it has the most resistant against Slow Read DoS Attack. The almost setting of attacker is the same as Section 3. 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 6) of the Slow Read DDoS Attack using these parameters. And from these settings, we can expect that the attack success status can be maintained from 6 to 60 seconds.

Table 6. Parameters of Attack Simulation

Parameters		Value
Apache Web server	Timeout	10
	ServerLimit	300
	MaxClients	300
Attackers	Connections Rate	50
	Number of Connections	1000

5 ATTACK SIMULATIONS

5.1 Outline of Attack Simulations

We set the parameters of attackers and Web server as same as in section 4.3, and they are summarized in Table 6. Before Slow Read DDoS Attack simulation, in order to analyze the effectiveness using huge number of attack connections, we do attack simulation by a single attacker with $N = 1{,}000$ (Simulation 1). Next, we simulate the Slow Read DDoS Attack by three attackers (Simulation 2) following the attack diagram shown in Figure 6. Each attacker's attack start time is set to $ta_1 = 0$ (sec), $ta_2 = 20$(sec) and $ta_3 = 40$ (sec) by Eq. (3).

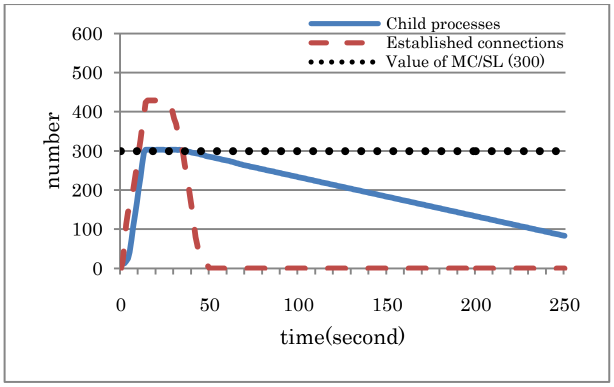

Figure 7. Simulation 1 (Attack result by At_1)

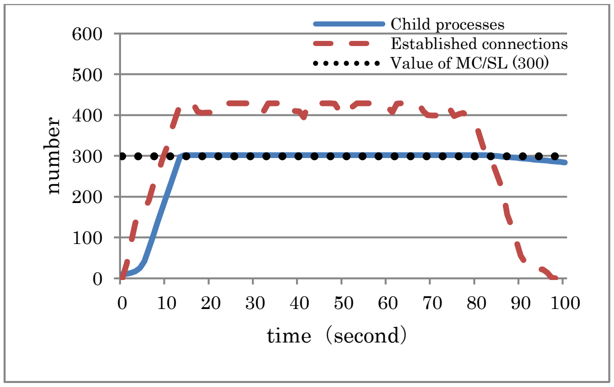

Figure 8. Simulation 2 (Attack result by At_1~At_3)

5.2 Results of Attack Simulations

Figure 7 shows the result of simulation 1. We can find longer attack success status (20 seconds) than the result of experiment 4 (8 seconds). We can confirm that the increasing N is adequate for improving attack result.

Figure 8 shows the result of simulation 2. We can find that the attack success status was maintained for 68 seconds (14~82seconds). It is longer than the theoretical attack diagram shown in Figure 6 (54 seconds). From this result, we can conclude that the countermeasure with short Timeout do not work against our new attack technique.

5.3 Consideration

From the simulation 1, we can consider that pending connections was also increased and 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 4.

In simulation 1, attack rate is 20.0 [second/one-attacker]. On the other hand, attack rate of simulation 2 is 22.7 [second/one-attacker]. As a result, we can find that Slow Read DDoS Attack is more efficient and lower-cost attack than simulation 1.

In simulation 2, we conduct the Slow Read DDoS Attack simulation by three attackers. As a result, attack success status was maintained longer than simulation 1. Therefore, we can consider that the attack success status can be maintained for a longer time, if three attackers attack repeatedly.

We can find two facts between the attack diagram (Figure 6) and the result of simulations (Figure 7 and 8).

1. *Time lag of child process generation*

From the result of attack shown in Figure 8, we can find that the attack starts to success after 14 seconds. On the other hand, from Figure 6, 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 computer generates. By predicting this time lag in advance, we will be able to set the attack start time to success exactly.

2. *Influence of the pending connections for establishment.*

Pending connections will be made, when the total number of attack connections (N) are set more than MC/SL. And they will be newly processed to establishment, when the time of Timeout passes. For this reason, we can consider that the pending connections make extension of the time length of attack success status. Thus, it was extended longer than the theoretical attack diagram (Figure 6).

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 above problems, we can realize more precise and lower-cost attack.

6 MODSECURITY AND ITS EFFECTIVENESS

6.1 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 the 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 [6]. The CRS provides configurable security rules such as follows [9].

- 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 IP address.

6.2 Experiment

6.2.1 Outline of Experiments

We set the attacker's parameters as same as Table 3 in sections 3.1. And we set the target Web server as same as experiment 4; 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 experiment to analyze the effectiveness of the Slow Read DoS Attack by a single attacker (Experiment 5). Next, we check the effectiveness of our proposal technique Slow Read DDoS Attack (Experiment 6) shown in section 4 with the same setting as simulation 2.

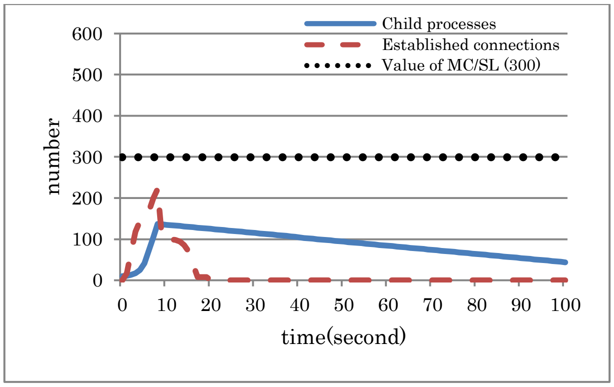

Figure 9. Experiment 5 (Timeout 10, MC/SL 300, Connection limit 100, Attack by At_1)

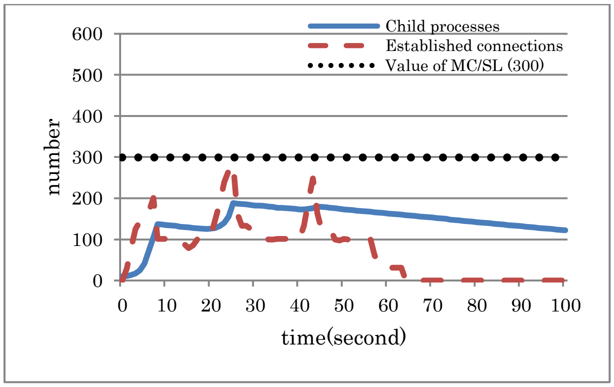

Figure 10. Experiment 6 (Timeout 10, MC/SL 300, Connection limit 100, Attack by At_1~At_3)

6.2.2 Results of Experiments

From Figure 9, we can see that ModSecurity functions after 9 seconds, because the number of established connections dramatically decreased. We can confirm that after ModSecurity starts, it can react immediately to the increase in attack connection. But since there is time lag of its starting, Web server allows many generations of child process which is more than the limitation number. As a result, 138 child processes are generated. Since the reduction in child process follows the setup of Timeout, in the case of experiment 5, child process decrease at rate 1[process/sec]. In the setting of experiment 5, we cannot success 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 10 shows the result of experiment 6. We can see that the attack also did not succeed at all with same reason 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 5. In addition, in the same way of experiment 5, by the time lag of ModSecurity starting, the total number of child processes increased more than 100 at 8 seconds and 25 seconds. This is in a situation as the purpose of Slow Read DDoS Attack.

However, in the sense of increasing the number of child process, contribution of At_1 is smaller than At_2. This is because it is set off against reduction in child processes generated by At_1. 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).

We experimented two types of heuristic attack settings in order to check our assumption. They are the condition 1 of $\mathrm{ta}_1 = 0, \mathrm{ta}_2 = 10, \mathrm{ta}_3 = 20$ and the condition 2 of $\mathrm{ta}_1 = 0, \mathrm{ta}_2 = 5, \mathrm{ta}_3 = 10$. Figure 11 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 cutting of attack connections by ModSecurity. As a result, the total number of child processes reached 293 at 25 seconds. Unfortunately, it has not yet resulted in the successful attack.

Figure 12 shows the result of condition 2. We can find that the attack success status is maintained for 18 seconds (18~36 seconds) and attack rate was 6.0 [second/one-attacker]. Therefore it has checked that our assumption was adequate. However, the attack diagram was derived heuristically from our attack results. 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. We can consider that this is because the pending connections which was waiting for establishment are newly processed as shown in the subject 2 of section 5.3.

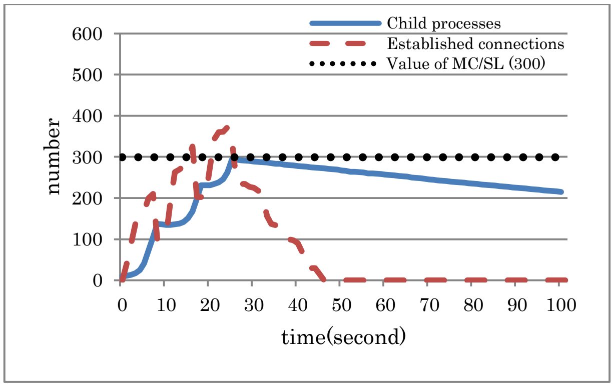

Figure 11. Condition 1($\mathrm{ta}_1 = 0, \mathrm{ta}_2 = 10, \mathrm{ta}_3 = 20$)

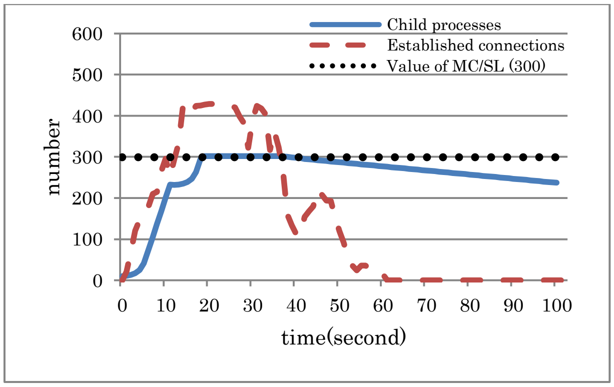

Figure 12. Condition 1($\mathrm{ta}_1 = 0, \mathrm{ta}_2 = 5, \mathrm{ta}_3 = 10$)

As the results, we can conclude that the security settings of Web server which applying the short 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 condition 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 7.

7 IMPROVEMENT OF THE SLOW READ DDOS ATTACK

7.1 Attack Strategy

In this section, we consider how to deceive ModSecirity's limitation. In the simple way, we take the technique of 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 effectivity of attack easily applying the technique shown in section 4.3 to attackers who divided into same groups. This is more positive than heuristic type as shown in above section. On the other hand, if the settings of limitation number in ModSecurity CRS are unknown, the number of attackers who should collect is not decided. So, it is necessary to perform the attacks such as experiment 5 in advance 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 [10]. So, it is easily considered from the above discussions to hold effective improved Slow Read DDoS Attack conditions.

In the followings, we assume that the setting of Web server is known to attacker and is the same as experiment 5 in section 6.2. 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 need to consist of at least three attackers (300/100=3). 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 Fig. 6. And the attackers of each group attack simultaneously. In other words, in order to ignore the function of ModSecurity, we increase the number of attackers to attack at the same time. Moreover, in order to maintain the attack success status for a long time, we have to increase attack groups.

7.2 Attack Simulation 3 and 4

7.2.1 Outline of Attack Simulations

We set the parameters of Web server and attacker as same as experiment 5. We conducted two types of attack simulation; simulation 3 and simulation 4. We assumed that three attack groups and one group consist of three attackers as shown in section 7.1. In these 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 Fig. 6. 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.

Simulation 3 denotes the minimum attack unit. From simulation 2 (see Figure 8), we obtained the successful attack result near theoretical estimation under the condition of $\mathrm{N_i} = 1{,}000$. Since 1,000 attack connections from one group are necessary, 340 attack connections per attacker are assigned. So, in the simulation 3, we set the attack condition with $\mathrm{N}_{11} \sim \mathrm{N}_{33} = 340$ to expect the effectiveness such as condition 1 shown in section 6.2.

Simulation 4 denotes a generous attack condition. The result of simulation 2 shows that the condition of $\mathrm{N_i} = 1{,}000.$ 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 the simulation 4, we set the attack condition with $\mathrm{N}_{11} \sim \mathrm{N}_{33} = 1{,}000$.

7.2.2 Results of Simulations

Figure 13 shows the result of simulation 3. We can find that the attack success status was maintained for total 54 seconds (13~63 seconds and 69~73 seconds). At 63 seconds, the number of child processes become lesser than 300 because of Timeout and ModSecurity, and cha-

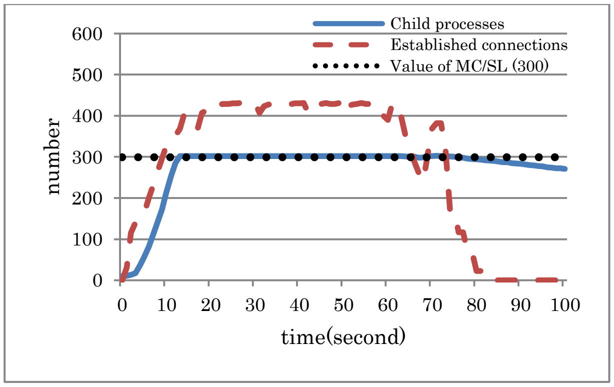

Figure 13. Simulation 3 (N_{11}~ N_{33} = 340)

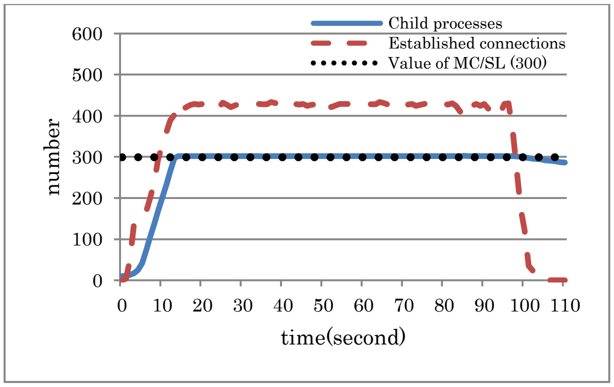

Figure 14. Simulation 4 (N_{11}~ N_{33} = 1,000)

nged to attack failure state. After 69 seconds, it returned to attack success state because of the effectiveness of Atg_3's established connections. However, as the same situation as condition 1 shown in Figure 11, the contribution of Atg_3 is lesser than Atg_2 . So, it maintained attack success state for only four seconds. As the result, we can judge that simulation 3 is succeeded in the attack.

Figure14 shows the result of simulation 4. On the whole, this simulation succeeded in the attack. 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 IP address can be ignored by our improved new attack technique.

7.3 Consideration

From the result of simulation 3 and simulation 4, the attack rate of simulation 3 is 6.0 [second/one-attacker] and of simulation 4 is 9.2 [second/one-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 7.1, in order to hold attack success status longer than simulation 4, we need to prepare other attack groups. 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 using ModSecurity.

Also in simulation 3 and simulation 4, we can conclude that the pending connection is important factor. We already pointed out this issue in section 5.3, analysis of the pending connection processing is useful for improvement of attack technique or defense strategy.

8 DISCUSSIONS

The important feature of Slow Read DoS Attack is that the target Web server is not out. From our attack simulations, it is clear from the fact that the target Web server certainly returned to the service available state after the attack. Conversely, it is difficult for administrators to detect the attack. Since especially 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 number of generated child process in a second is 32. Thus, under the condition of C=50, 18 attack connections are processed as pending connections. Since child process is valid within the period decided by Timeout, if we can control the rate of generating pending connection by C , we will be able to 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 becomes a cause by which the attack is easy to be detected. 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 real threat.

As already we described above, the construction of IDS against Slow Read DoS / 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. This is also our future work.

9 CONCLUSION

In this paper, we analyzed the effectiveness of Slow Read DoS Attack by computer simulations. From results, we concluded that the attack by a single attacker is not so efficient and it does not become real threat. However, from results, we can derive the improvement of the 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 simulation against the Web server with secure module, ModSecurity. ModSecurity can limit the period 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 8. And we concluded that the analysis of generation of pending connections using the time lag of starting security modules or child process is important factor to improve of the attack technique and development of countermeasures.

Note that our attack simulations are done in only one-hop virtual environment in order to be easy to analyze (see Figure 1). Therefore, the attack will effect 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 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 at less cost. The analysis of more actually based threat is also our future work.

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, Nos. 3/4, pp. 300-319, 2013

[2] Sergey Shekyan, “Are you ready for slow reading?,” https://community.qualys.com/blogs/securitylabs/2012/01/05/slow-read

[3] Kelly Jackson Higgins, “New Denial-Of-Service Attack Cripples Web Server By Reading Slowly, http://www.darkreading.com/attacks-breaches/new-denial-of-service-attack-cripples-we/232301367

[4] Sergey Shekyan, “slowhttptest,” https://code.google.com/p/slowhttptest (last access : July 23, 2014)

[5] W3Techs, “Most popular web servers,” http://w3techs.com

[6] ModSecurity, http://www.modsecurity.org

[7] ExtremeTech, “Virtual Machines & VMware Part II,” http://www.extremetech.com/computing/72268-virtual-machines-vmware-part-ii

[8] Wireshark, http://www.wireshark.org

[9] OWASP ModSecurity Core Rule Set Project, https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project

[10] 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.

[11] Cambiaso Enrico, Papaleo Gianluca, Chiola Giovanni and Aiello Maurizio, “Taxonomy of Slow DoS Attacks to Web Applications,” SNDS 2012, CCIS 335, pp. 195-204, 2012.

[12] Hiroshi, Kurakami, “The advanced DDoS attack and countermeasure,” Information Processing, Vol.54, No.5, pp. 475-480, 2013.

[13] Stephen M. Specht and Ruby B. 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

[14] KISA (Korea Internet Security Agency), The response guide against DDoS attack, https://www.boho.or.kr/kor/data/technicalList.jsp

[15] Ronen Kenig, “Why Low & Slow DDoS Application Attacks are Difficult to Mitigate,” http://blog.radware.com/security/2013/06

[16] ha.ckers, Slowloris HTTP DoS, http://ha.ckers.org/slowloris

[17] VMware Player, http://www.vmware.com/jp/products/player

[18] Apache, http://httpd.apache.org