# Slow Read DoS 攻撃とその対策に関する解析

朴 駿漢 †　　岩井 啓輔 †　　田中 秀磨 †　　黒川 恭一 †

† 防衛大学校
239-8686　神奈川県横須賀市走水 1-10-20
junhanp78@gmail.com, {iwai, hidema, kuro}@nda.ac.jp

**あらまし**　DoS/DDoS 攻撃によるサイバー攻撃は絶えず続いており，その手法もますます多様化，巧妙化している．本研究では，このような多様化した DoS 攻撃の 1 つである Slow Read DoS 攻撃に注目し，仮想環境を構築して Slow Read DoS 攻撃効果の解析を行った．その結果，単独の攻撃者だけで攻撃を行った場合，Web サーバでのセキュリティ設定及び同一 IP アドレスからの多数のリクエストを制限するセキュリティ対策を施すと攻撃を防ぐことができることが分かった．しかしながら，本論文で提案する複数の攻撃者が結託した Slow Read DDoS 攻撃で計算機実験による評価を行った結果，前述したセキュリティ対策だけでは Web サーバを攻撃から防ぐことが難しく，効率的に攻撃を実行できることが分かった．

# Analysis of Slow Read DoS Attack and Countermeasures

Junhan Park†　　Keisuke Iwai†　　Hidema Tanaka†　　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**　In our research, we focus on a Slow Read DoS attack which is one of sophisticated DoS techniques. In this paper, we analyze the effectiveness of Slow Read DoS attack using the virtual environment. As a result, we found 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 found that these countermeasures are not effective against Slow Read DDoS attack which is proposed in this paper.

## 1　はじめに

最近は攻撃者が大量のコネクションを Web サーバに張り、サーバのリソースを枯渇させる Slow HTTP 攻撃が増加しつつある [1], [2]．その中で 2012 年 1 月米国 Qualys 考案した Slow Read DoS 攻撃という手法に注目する．この手法は，攻撃クライアントが正当なリクエストを送信するが，Web サーバからのレスポンスを読み取る時間を引き延ばすと共に多くのコネクションを接続した状態に維持する．その結果，Web サーバが処理できるすべてのプロセスを使い切ることになり，他の正当なクライアントからの接続ができなくなる．また，この攻撃は正常のリクエストを行うため，検知するためにはネットワークレイヤをモニタリングしない限り難しい [3].

本研究では仮想環境を構築して Slow Read DoS 攻撃効果の解析を行った．解析は，テストツールである *slowhttptest* を用いた [4]．攻撃対象は，*Apache* Web サーバとした [5], [6]．解析の結果，単独の攻撃者による攻撃効果は制限的であり，Web サーバの Timeout の設定値により効果が決定されることが分かった．また，複数の攻撃者が結託した新たな攻撃手法である “Slow Read DDoS 攻撃” を提案し、攻撃効果が向上することを確認した．

さらに，同一 IP アドレスからの大量のリクエストを制限する ModSecurity を適用した Web サーバに

対しても攻撃効果を解析した．その結果，Slow Read DoS 攻撃は，完全に防ぐことができた．しかしながら，提案した手法で攻撃すると ModSecurity のセキュリティ機能を克服することができることを計算機実験で確認した．このようなセキュリティ対策は，攻撃成功の期間を短縮することはできるが，攻撃を根本的防御することはできないことが分かった．攻撃実験結果から，子プロセスの生成手順及びセキュリティモジュール起動のタイムラグまでの解析を行うことでさらなる効率性の向上を見込めると予想される．また，防御的な観点からもこれらの解析は重要であり，さらに IDS との関係性について考察を示す．

## 2　攻撃効果の解析

### 2.1　実験環境

実験環境を図 1 に示す．Web ページのサイズを 100KB とし，*Wireshark*[9] を用いて仮想 switch (VMnet8) を観測した．攻撃対象となる Apache Web サーバの httpd.conf 中，クライアントとのコネクションを制御する Directive の設定を表 1 に示す．子プロセスの生成を制御する prefork MPM(Multi Processing Module) の設定を表 2 に，slowhttptest ツールの攻撃オプションの値を表 3 に示す．

### 2.2　パラメータの機能

Slow Read DoS 攻撃を行った場合，サービス不能状態になるのは，攻撃コネクションが Max Clients(以下 MC) 値以上に接続される時である．しかし，設定した Timeout 値が経過すると攻撃コネクションは強制的に切断され，サービス可能状態に戻る．このように，Timeout の設定は，攻撃コネクションと Web サーバの接続を制御する．また，MC 値は，コネクション数の上限である．もし、攻撃コネクション数が MC 値より多ければ全ての子プロセスを使い切ることになり，サービス不能状態になるが MC 値より少なければサービス不能状態にならない．Sever Limit(以下 SL) の値は，起動可能な子プロセス数を制御するので MC 値と同じ値で設定した．以下では，MC/SL と併記する．ただし，攻撃の設定は表 3 のように固定した．

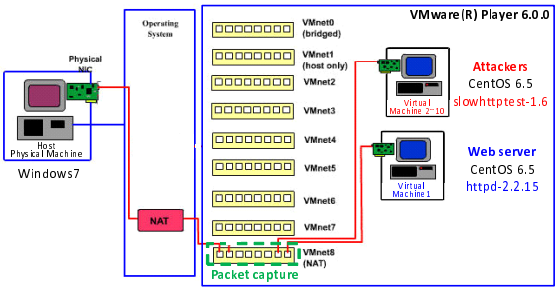


図 1: 実験環境 [8]

表 1: Directive

| Directive | 値 |
|---|---|
| Timeout | 60 |
| KeepAlive | Off |

表 2: Prefork MPM

| Directive | 値 |
|---|---|
| StartSevers | 8 |
| MinSpareServers | 5 |
| MaxSpareServers | 20 |
| ServerLimit | 256 |
| MaxClients | 256 |
| MaxRequestChild | 4000 |

表 3: Slow Read DoS 攻撃オプション

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

### 2.3　用語定義

Slow Read DoS 攻撃実験を行う際，攻撃に成功したか否かを判定する用語について定義する．“攻撃成功” は，MC/SL で設定した値以上のコネクション数を接続し，強制的に切断される前までその接続が一定時間維持することを示す．すなわち，MC/SL 値以上の攻撃コネクション接続数の場合，これに当該する．“攻撃失敗” は，MC/SL 値未満の攻撃コネクション接続数の場合であり，Web サーバが他の正当な接続を許すサービス可能状態を示す．従って，攻撃

の効果はサービス不能を維持した期間 (秒) で判断できる．その他，確立済みのコネクションとは，TCP の 3-way-handshake により確立されているが，Web サーバの同時接続数が満杯になり処理できない状態のコネクションである．確立待ちのコネクションとは，TCP の 3-way-handshake による確立を待っているクライアントからのコネクション (SYN パケット) である．これらを合わせて保留中のコネクション (pending connection) と呼ぶ．

# 3 Slow Read DDoS 攻撃

## 3.1 手法の概要

SCIS2014 の結果で Timeout が経過すると，接続されていたコネクションが強制的に切断され，サービス不能状態からサービス可能状態に復帰することを確認した．さらに，Timeout 値を 10 秒と設定することにより，攻撃成功期間が極端に短くなり，攻撃効果を軽減することが可能であることが分かった．また，MC/SL の値未満の攻撃コネクション数では，必ず攻撃が成功しないことが分かった．すなわち，単独の攻撃者では Web サーバをサービス不能状態に維持させるのは限定的であり，Slow Read DoS 攻撃の限界が明らかになったと言える．逆に言えば，Web サーバの Timeout 設定値により攻撃コネクションが切断される前に，別の攻撃者が新しい攻撃コネクションを発行開始すれば，サービス不能状態を効率的に維持できると予想される．本論文では，この攻撃手法を “Slow Read DDoS 攻撃” と呼ぶ [10]．

## 3.2 攻撃成功状態になる条件

SCIS2014 の実験結果により，攻撃成功条件をモデル化する．表 4 に使用する変数を示す．攻撃成功状態である時，攻撃者が張ったコネクション総数を A とすれば，$M \leq A$ が満たされる．$A$ は，$t_0$ と $t_z$ の時間から 2 通りの算出ができる．$t_0 \geq t_z$ の場合，

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

$t_0 < t_z$ の場合，

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

ここで，$t \geq 0$ 秒であり，攻撃開始後の時間の流れを表す．

表 4: 変数

| 変数 | 説明 |
|---|---|
| $t_0$ | Timeout 設定値 |
| $t_z$ | 攻撃コネクションを張り終える時刻 ($t_z = N/C$) |
| $N$ | 攻撃コネクション総数 |
| $C$ | 1 秒当たり張る攻撃コネクション数 |
| $K$ | 時刻 $t_0$ 後に、1 秒当たりに切断されるコネクション数 |
| $M$ | サーバが処理できる総コネクション数 (MC/SL) |
| $A_{(t)}$ | 時刻 $t$ 秒までに攻撃者が張ったコネクションの総数 |

## 3.3 提案手法

式 (1) と (2) を満たすために，複数の攻撃者による手法を考察する．簡単のために，ここでは，式 (1) である $t_0 \geq t_z$ の場合を示す．この場合，$t \geq t_0$ の時，式 (2) の $t \geq t_0$ と同様の状態になる．基本方針は第 1 の攻撃者が攻撃コネクションを張り終える時刻 $t_z$ 以前に第 2 の攻撃者が攻撃コネクションを張り始めれば良い．同様にこれを繰り返すことで，サービス不能状態を維持できると考えられる．$At_1$，$At_2$，…，$At_N$ の $N$ 人の攻撃者を仮定した時，$n(2 \leq n \leq N)$ 番目の攻撃者 $At_n$ が攻撃コネクションを張るべき時刻 $ta_n$ は以下のように書ける．

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

ここで $C_i$ と $N_i$ は，それぞれ攻撃者 $At_i$ の設定した $C$ 及び $N$ の値である．例として，3 人の攻撃者による Slow Read DDoS 攻撃の理論的な攻撃ダイヤグラムを図 2 に示す．ここでは SCIS2014 で示した実験結果のうち，最も攻撃耐性があったサーバの設定を攻撃目標と仮定している (Timeout 10, MC/SL 300)．また，攻撃者の設定は，表 3 の設定と同様に 1 秒当たりに張る攻撃コネクション数 $C = 50$ とする．ただし，Timeout の設定が 10(秒) なので，攻撃コネクション総数 $N \geq C \times 10$ 以上にしなければ攻撃が成立しない．ここでは $N = 1000$ とし，$t_z = 20$(秒) の設定とする．従って，$C_1 = C_2 = C_3 = 50$，$N_1 = N_2 = N_3 = 1000$，$t_0 = 10$(秒)，$M = 300$ とした．図 2 により，6~60 秒 (54 秒間) の攻撃成功

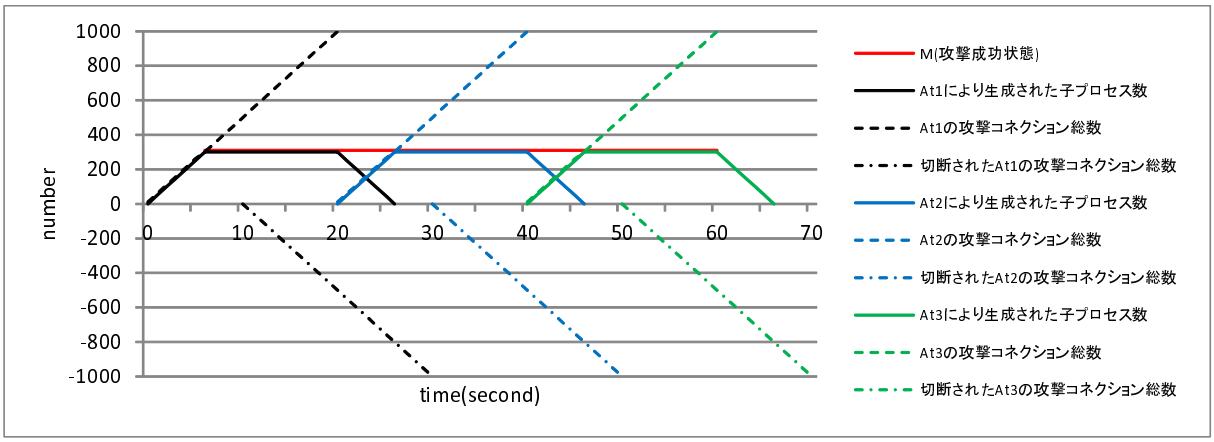


図 2: 攻撃ダイヤグラム

表 5: 計算機実験のパラメータ

| パラメータ | | 値 |
|---|---|---|
| Web サーバ | Timeout | 10 |
| | ServerLimit | 300 |
| | MaxClients | 300 |
| 攻撃者 | Connections Rate | 50 |
| | Number of Attack connections | 1000 |

状態を維持できると予想できる．このように攻撃対象の設定値，参加する攻撃者の数とそれぞれのパラメータ値を決定し，サービス不能維持期間を考慮すれば Slow Read DDoS 攻撃の攻撃ダイヤグラムを作成できる．この攻撃ダイヤグラムに従って攻撃を実行すれば，攻撃対象 Web サーバは設定した期間，サービス不能に陥る．

### 3.4 計算機実験による評価 (実験 1)

計算機実験における攻撃者の設定及び Web サーバの設定は 3.3 節と同様とし，表 5 にまとめて示す．3 人の攻撃者 $At_1$〜$At_3$ による攻撃 (実験 1) の結果を図 3 に示す．この結果から攻撃成功期間が 14〜82 秒の 68 秒間維持されたことが分かる．この結果は，3.3 節で示した予想を上回っている．

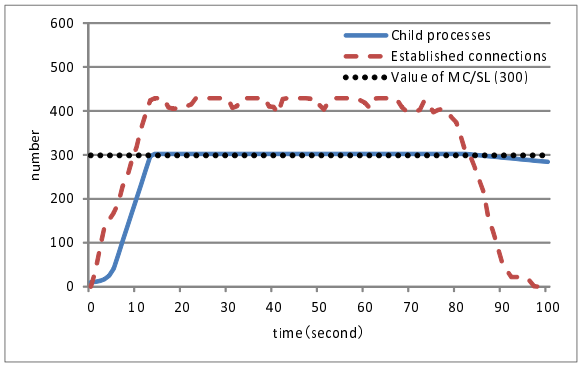


図 3: 攻撃者 $At_1$〜$At_3$ の攻撃結果 (実験 1)

## 4 ModSecurity とその効果

### 4.1 ModSecurity

ModSecurity は，WAF (Web Application Firewall) の一つである．これは Apache Web サーバ，IIS，NGINX でリアルタイムで Web アプリケーションをモニタリング，ロギング，アクセス制御の機能を提供する．本研究では，OWASP (Open Web Application Security Project) が提供する ModSecurity CRS(Core Rule Set) を使用する．ここでは，ModSecurity を導入した Web サーバに対し，Slow Read DoS 攻撃効果を解析するために “HTTP Denial of Service Protections” の機能を用いる [11]．これは，一般的に多くの Web サーバが採用しているセキュリティ対策で，同一 IP アドレスからの大量のリクエストを制限する．

### 4.2 評価実験 (実験 2，実験 3)

#### 4.2.1 実験概要

実験における攻撃者の設定は，3.4 節の実験と同様である．Web サーバの設定は，Timeout 10 秒，MC/SL 300 に加えて ModSecurity を導入し，同一 IP アドレスからのリクエスト数を 100 まで制限す

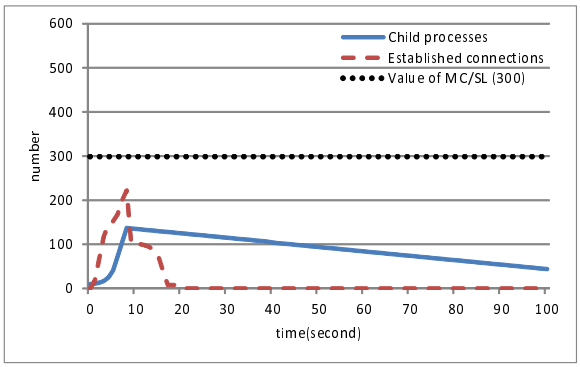


図 4: 攻撃者 $At_1$ の攻撃結果 (実験 2)

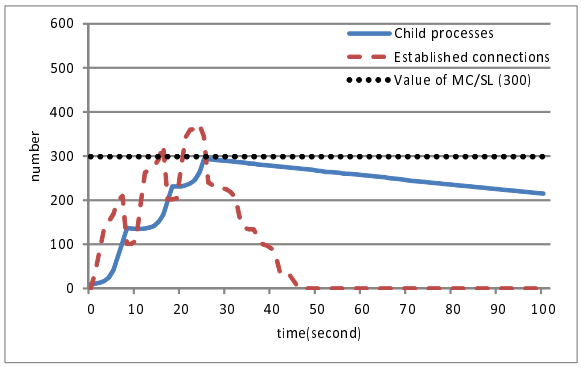


図 6: 条件 1($ta_1$=0, $ta_2$=10, $ta_3$=20)

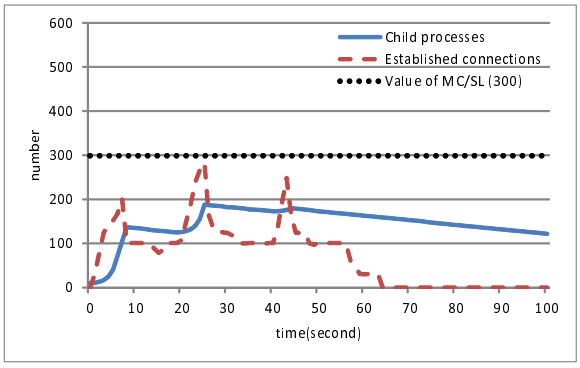


図 5: 攻撃者 $Atg_1$~$Atg_3$ の攻撃結果 (実験 3)

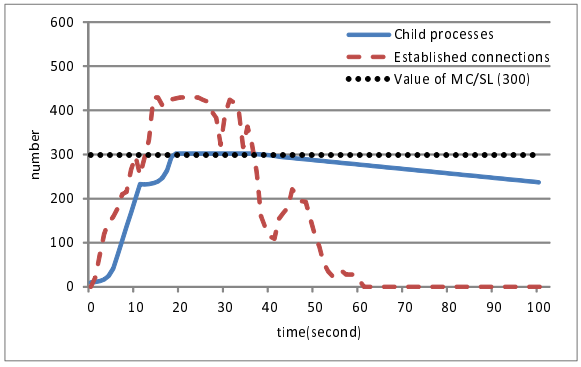


図 7: 条件 2($ta_1$=0, $ta_2$=5, $ta_3$=10)

る．まず，単独の攻撃者による Slow Read DoS 攻撃効果を解析するために実験を行う (実験 2)．次に，3 節で提案した Slow Read DDoS 攻撃の効果を確認するため実験 1 と同様の設定で実験を行う (実験 3)．

### 4.2.2 実験結果

実験 2 の結果を図 4 に示す．本結果は攻撃失敗であり，ModSecurity の効果が確認できた．また，図 4 から 9 秒後に TCP の確立数が急減しているので ModSecurity の機能が動作していることが分かる．しかし，その機能が起動するのに若干のタイムラグがあるので Web サーバは制限数より多くの子プロセスの生成を許している．その結果，138 の子プロセスが生成された．結果的には，ModSecurity は単独の攻撃者による Slow Read DoS 攻撃に対して十分な効果があると言える．

実験 3 の結果を図 5 に示す．ここでも上述した理由で攻撃が成功しなかったことが確認できる．TCP の確立数の時間変移から，実験 2 と同様に ModSecurity は十分な効果があることが確認できる．また，ModSecurity の動作のタイムラグにより子プロセスの総数が 8 秒 ~25 秒間 100 以上増加された．これは提案した Slow Read DDoS 攻撃の目的としての状況である．しかしながら，子プロセスの意味において $At_3$ の攻撃効果は $At_2$ より小さい．これは $At_1$ によって生成される子プロセスの縮小に対し相殺されるからである．そこで式 (3) による各攻撃者攻撃開始時刻 ($ta_n$) の計算結果よりその間隔を短くすれば攻撃が成功できると考えられる．

この予想を確認するため 2 種類のヒューリスティックな攻撃設定の実験を行った．まず条件 1 として $ta_1$=0, $ta_2$=10, $ta_3$=20 と設定し，条件 2 として $ta_1$=0, $ta_2$=5, $ta_3$=10 と設定した．条件 1 の結果を図 6 に示す．この結果から，ModSecurity による攻撃コネクションの切断及び短い Timeout による減少の影響を受けずに総子プロセス数が増加していることが確認できる．その結果，総子プロセス数が 25 秒に 293 まで到達したが攻撃成功には至らなかった．

条件 2 の結果を図 7 に示す．この結果から，18 秒間 (18~36 秒) 攻撃が成功し，攻撃レートが 6.0[second/one-attacker] であることが分かる．従って上述した

推定が正しいことが確認できた．また，31 秒と 45 秒に TCP の確立数が増加していることが分かる．これは SCIS2014[10] で述べたように確立待ちのコネクションが新たに処理されたからであると考えられる．

これらの結果から，短い Timeout を設定し，ModSecurity を導入した Web サーバは，単独の攻撃者からの Slow Read DoS 攻撃及び単純な Slow Read DDoS 攻撃に対し十分な防衛効果があると言える．しかしながら，条件 1 及び条件 2 の結果から，このような対策の防御効果を軽減することができると考えられる．

# 5 Slow Read DDoS 攻撃の改良

## 5.1 攻撃戦略

本節では，攻撃者数を増加することにより同一 IP アドレスからの接続数を縮小する手法を考える．これは複数の攻撃グループが 3.3 節で示した手法で攻撃することである．この手法の利点は攻撃の有効性を容易に予測することができることである．一方で，ModSecurity の制限数の設定が分からないと，グループ毎に集めなければならない攻撃者数は決められない．そこで予め実験 2 のように攻撃を行い，予測する必要がある．ここでは攻撃者が 4.2 節の実験 2 と同様に Web サーバの設定を知っていると仮定する．ModSecurity が制限している同じ IP アドレスからの同時接続数は 100 と設定されているので，攻撃者一人が生成できる攻撃コネクションは 100 である．MC/SL は 300 であるため，一つのグループが必要な攻撃者の数は最低 3 人である (300/100=3)．従って，次の構成は，最小攻撃グループ構成である．

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

最小攻撃グループは，ModSecurity の制限数と MC/SL 値から容易に構成できる．各攻撃グループは図 2 の攻撃ダイヤグラムに従い逐次的に攻撃を行う．ただし，各攻撃グループの攻撃者は同時に攻撃をする．つまり，ModSecurity の機能を克服するために同時に攻撃する攻撃者を増やす戦略である．さらに，攻撃成功状態をより長く維持させるためには攻撃グループの数を増やせば実現できる．

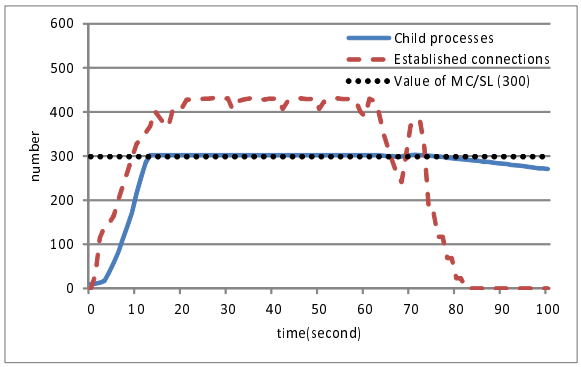


図 8: $N_{11}$∼$N_{33}$=340 の攻撃結果 (実験 4)

## 5.2 評価実験 (実験 4, 実験 5)

### 5.2.1 実験概要

Web サーバと攻撃者の設定は実験 2 と同様にした．また 5.1 節で示したように 1 つのグループに攻撃者 3 人が構成されている 3 つの攻撃グループを想定した．攻撃ダイヤグラムは $tg_1$=0(秒), $tg_2$=20(秒), $tg_3$=40(秒) とした．

実験 4 は，最小攻撃グループ構成の攻撃解析を示す．実験 1 から，$N_i$=1,000 の条件で理論的な評価に近い攻撃成功を得たので，一人あたり 340 の攻撃コネクションを設定する．

実験 5 は，余裕のある条件の攻撃解析を示す．実験 1 の結果は，$N_i$=1,000 が各攻撃者にとって高くないコストであることを示している．この実験の目的は保留中のコネクションの生成が攻撃結果にどのように影響するかを解析することである．従って，実験 5 では，総攻撃コネクション数は，$N_{11}$∼$N_{33}$=1,000 と設定した．

### 5.2.2 実験結果

図 8 に実験 4 の結果を示す．総攻撃成功状態は 54 秒間 (13∼63 秒及び 69∼73 秒) 維持できたことが分かる．63 秒に Web サーバの Timeout の設定と ModSecurity の制限で子プロセスの数が 300 以下になり，攻撃失敗状態になった．しかしながら，69 秒後には $Atg_3$ の保留中のコネクションの影響により攻撃成功状態に復帰した．これは図 6 の条件 1 と同様に $Atg_3$ の寄与が $Atg_2$ より小さいので，わずか 4 秒間のみ攻撃成功状態が維持されたと考えられる．

図 9 は，実験 5 の結果を示す．全体的に攻撃成功していることが分かる．実験 1 と比べると 1 つの

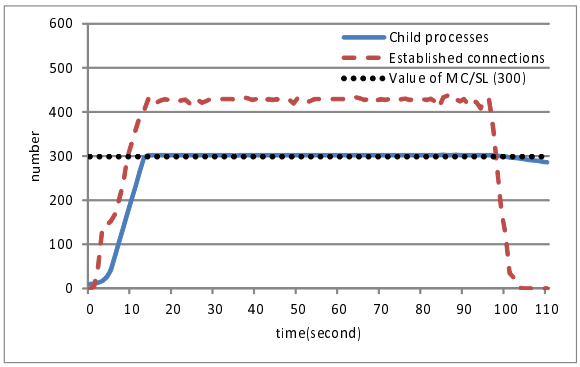


図 9: $N_{11}$∼$N_{33}$=1,000 の攻撃結果 (実験 5)

グループを 1 人の攻撃者であると仮定すると実験 5 は攻撃コネクション総数が実験 1 より 3 倍の結果である．このような観点から実験 5 の条件の場合，ModSecurity を適用しているが結果的に実験 1 より効率的ではないと言える．より効率的に $N$ を決める方法があると考えられので，これに関しては今後の課題としたい．

これらの結果から，改良した新しい攻撃手法は同一 IP アドレスからのコネクション制限を克服することができると結論できる．

### 5.3 考察

実験 4 と実験 5 の結果から実験 4 の攻撃レートは 6.0[second/one-attacker] で，実験 5 は，9.2[second/one-attacker] である．実験 1 と比べると Web サーバに ModSecurity を適用することに伴い，攻撃コストが上がった．この観点では ModSecurity は対策として効果的であると言える．しかしながら，ModSecurity は攻撃を完全に防ぐことができず，“HTTP Denial of Service Protections” は，改良した Slow Read DDoS 攻撃に対しては十分な有効性を持ってない．

5.1 節で示したように，実験 5 より攻撃成功状態を長くさせるためには新規の攻撃グループを準備する必要がある．一方で，ModSecurity を使用していない場合は理論的には二人の攻撃者が交互で攻撃を繰り返すことで永遠に攻撃成功状態を維持することができる．ModSecurity を使用する場合は，ボットネットを利用し，改良した Slow Read DDoS 攻撃を行なっても攻撃成功状態の期間は有限である．これが ModSecurity が対策として使われる観点で重大なポイントである．

## 6 おわりに

Slow Read DoS 攻撃の重要な特徴はターゲットの Web サーバが落ちないことである．今までの攻撃実験から，攻撃が終わると Web サーバはサービスが可能な状態に戻っている．逆に言えばこのような状況は Web サーバの管理者が攻撃と検知することが難しい．特に正当なリクエストを送っているのでシグネチャータイプの IDS ではこの攻撃を検知することが不可能であると予想される．

今回の攻撃実験では攻撃戦略を単純化するために次のような条件を与えた．

1. $C$=50 で固定．
2. window size は 0 に固定．

ターゲットが Apache であるときは，1 秒あたり生成される子プロセスの数は 32 個である．したがって，$C$=50 で攻撃をすると 18 個の攻撃コネクションは保留中のコネクションとして処理される．子プロセスはタイムアウトにより決定された期間内のみ有効であるので、もし $C$ による保留中のコネクションの生成レートをコントロールすることができるとより効率的な攻撃手法を開発することができる．

本研究での攻撃シナリオで攻撃者は，“window size = 0” とした．しかしながら，実際に攻撃する場合は，この条件は必要ない．実際、これは攻撃検知を容易にする．従って，window size の値は，必要最小限度に設定すればよい．その結果、攻撃の有効性は今回の攻撃実験より劣るが，攻撃検知の危険は小さくなる．さらに，window size がセッション毎に柔軟に変更する手法も考えられる．このように，$C$ 及び window size の値が変更される適応的な Slow Read DoS の評価は今後の課題である．

また，ターゲットの Web サーバが ModSecurity を使用しても改良 Slow Read DDoS 攻撃に対しては防御することが難しいことを示した．この攻撃は組織化した攻撃グループが必要である．しかしながら，最近のサイバー攻撃及びサイバー犯罪では組織化されたグループによる実行は一般的なことである．このような攻撃シナリオは現実的であると認識すべきである．

前述したように，本攻撃に対する IDS の構築は非常に難しい．しかしながら，適応的な攻撃はいくつかの特徴を持つと考えられる．したがって，アノマ

リータイプの IDS はこのような特徴の発見により構築可能と考えられる．これも今後の課題としたい．

## 参考文献


[1] Cambiaso, E., Papaleo, G., Chiola, G. and Aiello, M., “ Slow DoS attacks: definition and categorisation”, Int. J. Trust Management in Computing and Communications, Volume 1, Number 3-4 pp.300-319, 2013.

[2] キーマンズネット, “安心安全な Web サイトの作り方 〜スロークライアントアタック〜” http://www.keyman.or.jp/kc/sec/firewall/30006788/, 2013

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

[4] slowhttptest, https://code.google.com/p/slowhttptest/, 2014

[5] w3techs, “Most popular web servers” http://w3techs.com/, 2014

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

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

[8] ExtremeTech, “Virtual Machines and VMware Part II” http://www.extremetech.com

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

[10] 朴 駿漢，岩井 啓輔，田中 秀磨，黒川 恭一，“Web サーバへの Slow Read DoS 攻撃に関する考察”， SCIS2014.

[11] OWASP ModSecurity Core Rule Set, http://spiderlabs.github.io/owasp-modsecurity-crs

[12] Esraa, A., B. B. Gupta, Shankar, K.: 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, 24–32 (2012)

[13] 倉上 弘, “高度化する DDoS 攻撃と対策サイトの視点から”, 情報処理, Vol.54, No.5, pp.475-480, 2013.

[14] 情報処理推進機構, “「サービス妨害攻撃の対策等調査」報告書” http://www.ipa.go.jp/security/fy22/reports/isec-dos/index.html, 2010

[15] 鶴長 鎮一, “「サーバ構築の実際がわかる Apache[実践] 運用/ 管理」”, 技術評論社, 2012.

[16] Richard Stevens, “UNIX NETWORK PROGRAMMING”, Vol.1, 第 2 版, ピアソンエデュケーション, 1999.

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

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

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

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