DNSを用いたDDoS攻撃回避システム

阿部幸太朗* 森口一郎**

DDoS攻撃による被害軽減を目的としたシステムの構築と評価を行った。DDoS攻撃の対策としてはトラフィック量を制御することが一般的であるが、この手法では正規ユーザのパケットもトラフィック量の中に含まれるため正規ユーザだけにサービスを提供することができない。本システムでは、まずIDSが攻撃ホストからの攻撃を検知し、Webサーバ及びDNSサーバにDDoS攻撃回避要求を申請する。回避要求を受けたWebサーバは自身のIP addressを変更することで攻撃を回避する。また、正規ユーザを回避先のWebサーバに誘導するため、DNSサーバはzoneファイルのAレコードを変更する。しかし、Webサーバの IP addressを変更する際、ネットワークを再起動する必要があるため、IP address切り替え時間を考慮しWebサーバを２台稼動させたdual apacheシステムを構築し、性能比較を行った。この結果、DDoS攻撃に対して本システムの有効性を示すことができた。しかし、IP addressを変更しても再びDNSに対し正引きアクセスする機能を持つ攻撃に対しては、本システムが性能を十分に発揮できないことも明らかになった。

キーワード：DNS、DDoS攻撃、IDS、ネットワークセキュリティ

DDoS Attack Evading System by DNS

Kohtarou ABE* and Ichirou MORIGUCHI**

In this study, damage reducing system against DDoS was developed and evaluated. Although generally controlling the amount of traffic is a most common method, packets of normal users are also affected by this strategy. Firstly a intrusion detection system detects a packet of DDoS, and instructs a Web server and a DNS server to evade the DDoS attack. After receiving the request, the Web server changes its IP address to avoid the attack. Simultaneous the DNS server rewrites the address record in zone file to lead normal users to the new address of the Web server. However, because it is necessary to reboot the network function of Web server for changing the IP address, a new system（dual apache system）that comprises two Web servers for the purpose of reducing the time if rebooting network was also built and compared with the system including one Web server. As a result this proposed DDoS evading system confirmed its efficiency against DDoS attacks. It, however, was also found that the system of this study is not so effective against the DDoS attack tools which repeatedly access DNS server for name resolutions.

Keywords: dns, ddos attack, ids, network security

*東京情報大学　総合情報学部　情報システム学科学部学生
Tokyo University of Information Sciences, Faculty of Informatics, Department of Information Systems
**東京情報大学　総合情報学部　情報システム学科
Tokyo University of Information Sciences, Faculty of Informatics, Department of Information Systems

2013年12月10日受理

1．はじめに

情報化社会の発展に伴い、インターネットを利用したサイバー攻撃が増大してきた。インターネット上で行われているサイバー攻撃の大半がDoS（Denial of Service）攻撃またはDDoS（Distributed Denial of Service）攻撃である[1]。DoS攻撃とは、サービス運用妨害攻撃と呼ばれるもので、サービスそのものを利用できなくする攻撃である。一方、DDoS攻撃は分散DoS攻撃と呼ばれ、複数台の攻撃マシンを準備し、そのマシンから攻撃を仕掛ける手法である。DDoS攻撃を対策するうえでトラフィック量を制御し対策することが一般的であるが[2]、この対策ではトラフィックの中に正規ユーザの通信パケットも含まれるため、正規ユーザもWebサーバにアクセスできないことになる。また、モバイルIPv6 技術を応用したIP addressを不定にすることによる攻撃回避方式を用いた標的サーバを守る研究が行われているが[3]、IPv6 の仕様に依存しているため、正規ユーザはMobile IPv6 に対応しなければならない。

本研究では、既存研究にはないWebサーバのIP addressを変更する手法を用いて、正規ユーザに影響を与えずかつIPのバージョンに依存しないDDoS攻撃による被害を軽減するシステムを構築した。このシステムは、攻撃を検知するIDSサーバ、正規ユーザにドメインからIP addressを渡すDNSサーバ、攻撃対象となるWebサーバの3種類のサーバから成り立っている。

本研究で構築した回避システムの流れとしては、攻撃ホストからWebサーバに攻撃が行われた際、まずIDSサーバが攻撃を検知し、DNSサーバ及びWebサーバに回避先IP addressを報告し、回避要請をする。次に、要請を受けたWebサーバはIP addressの変更を行うことにより、DDoS攻撃を回避する。IDSサーバからの要請に従い、DNSサーバはAレコードに書かれているWebサーバのIP addressの値を変更する。このシステムに正規ユーザがアクセスした際、正規ユーザはWebサーバにアクセスするためにDNSサーバに接続し、URLからIP addressを取得（正引きアクセス）する。取得したIP addressを元にWebサーバにアクセスすることによって回避先のIP addressに誘導され正規ユーザはコンテンツを取得することが可能となる。このことにより、正規ユーザをWebサーバに誘導でき、適切なサービスを提供できる。

今回はWebサーバをDDoS攻撃から回避するシステムの構築を行ったが、提供するサービスの種類に依存しないためあらゆるサーバを回避させることが可能である。

また、本システムの有効性を評価するため、攻撃ホストによるDDoS攻撃を行い、同時に正規ユーザを模してWebサーバにアクセスし、その際のアクセス成功率で本システムの有効性を評価した。その結果、DDoS攻撃に対し本システムの有効性を示すことができた。

2．システムの構成

本システムは、3種のサーバ（IDS, DNS, Web）で構築される（図1）。

攻撃ホストからWebサーバへDDoS攻撃が行われた際、はじめに、IDSサーバがruleファ

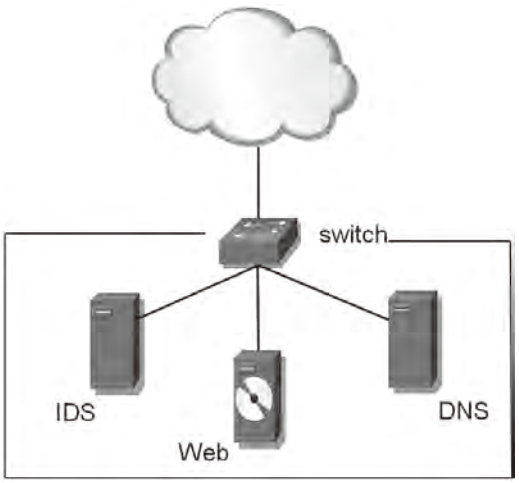

図1　DNSを用いたDDoS攻撃回避システム

イルを元に攻撃を検知し[4]、alertをIDSサーバ内の回避プログラムに出力する。次に、alertを出力された回避プログラムがDNSサーバ及びWebサーバに回避先IP addressを伝え、DDoS攻撃回避要請をする。さらに、要請を受けたWebサーバはIP addressを変更するscriptを実行することで、IP addressの変更を行い、DDoS攻撃を回避する。また、DNSサーバは、zoneファイルのAレコードのWebサーバのIP addressを変更する。

このシステムに対して正規ユーザは、DNSサーバに接続し、URLからIP addressを取得する。取得したIP addressを元にWebサーバにアクセスし、コンテンツを取得する。

攻撃ホストは、攻撃を開始する前に攻撃先のWebサーバのIP addressを取得するため、DNSサーバに接続し、URLからIP addressを取得（正引きアクセス）する。その後、攻撃ホストは取得したIP addressを元に攻撃を開始する。その攻撃をIDSサーバが検知しWebサーバおよびDNSサーバにDDoS攻撃回避要請をして、WebサーバのIP addressが変更されるため攻撃は失敗する。攻撃ホストによっては、変更されたIP addressを再度取得して攻撃を再開する場合もあるが、IP addressを変更することで必ず失敗する。しかし、正規ユーザは毎回DNSサーバに正引きアクセスを行うため、Webサーバへのアクセスが成功する。これらの正規ユーザと攻撃ホストの動作の違いにより正規ユーザだけにサービスを提供することが可能となる。ただし、正規ユーザがDNSキャッシュを持つ場合はWebサーバへのアクセスに失敗するが、これについては、対策方法も含めて以下で説明する。

今回構築したマシンのOSは全てVine5.2で、kernelのversionは2.6.27-67vl5を使用したが、どのLinux系OSでも本システムを構築可能である。実験環境で使用したIP addressは下記の通りである。

DDoS攻撃回避システム

IDSサーバ	192.168.136.43
DNSサーバ	192.168.136.15
Webサーバ	192.168.136.140～149

検証システム

攻撃ホスト	192.168.136.10
攻撃ホスト	192.168.136.30
正規ユーザ	192.168.136.20

本システムでは、正規ユーザがDNSキャッシュを持つことにより、正規ユーザがアドレス変更したWebサーバにアクセスできない問題があるため、DNSサーバを構築する際にこの問題を考慮しなければならない。この問題のメカニズムはまず、一般的な正規ユーザは一度Webサーバにアクセスした際、DNSサーバにアクセスしURLとIP addressを取得し、DNSキャッシュを保存する。二度目のアクセスの際は、保存したDNSキャッシュを用いDNSサーバにアクセスせずに直接Webサーバにアクセスする。しかし、本システムではIDSサーバが攻撃を検知した場合、DDoS攻撃回避要請によりWebサーバのIP addressが変更される。その場合、正規ユーザはDNSキャッシュを用いてアクセスするので、回避する前のIP addressに接続要求をしてしまい、アクセスすることができなくなってしまう。例えば、WebサーバのIP addressが192.168.136.140とする。正規ユーザの一度目のアクセスで、DNSサーバへ正引きアクセスを行いURLから192.168.136.140のIP addressを取得する。そのIP address（192.168.136.140）を用いてWebサーバにアクセスする（図２）。ここで、本システムが攻撃を検知し、DNSサーバのzoneファイルにあるAレコードのWebサーバのIP addressの値を192.168.136.141に変更し、WebサーバのIP addressを192.168.136.141に変更したとする。このタイミングで正規ユーザが２度目のアクセスを試みるとき、前回DNSサーバへ正引きアクセスし取得したIP address（192.168.136.140:

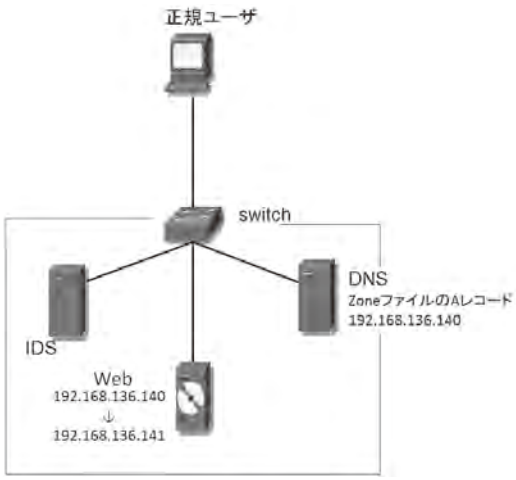

図２　DNSキャッシュ問題１

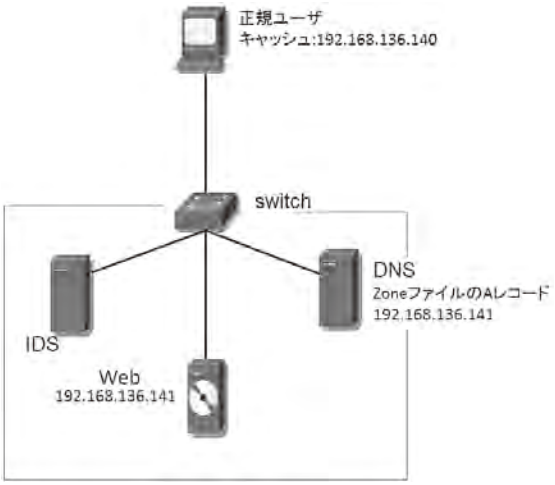

図３　DNSキャッシュ問題２

DNSキャッシュ）を用いてWebサーバにアクセスを試みる。しかし、Webサーバは192.168.136.141に変更しているのでアクセスが失敗してしまう（図３）。

このようなDNSキャッシュによる問題を解決するため、DNSのzoneファイルのTTLの値を低い値に設定した。

```
$TTL 1
```

zoneファイルにおけるTTLとは、ドメイン情報を参照してDNSキャッシュを保存する際に、保存する期間を指定する値のことである。この値を「１」に設定することにより、正規ユーザのDNSキャッシュの保持する時間を１秒に設定できる。つまり、１秒後には正規ユーザ側で自動的にDNSキャッシュを破棄するためDNSキャッシュを用いてアクセスすることがなく、正規ユーザが回避先のIP addressに正しくアクセスすることが可能となる。

しかし、正規ユーザにDNSキャッシュを持たせないため、Webサーバにアクセスを試みる度に毎回DNSサーバに正引きアクセスをする。このことにより、DNSサーバへのアクセス増加や、IP address取得時間に時間がかかり応答性が悪くなる問題が懸念される。しかし、正規ユーザはDNSサーバに一つのUDPパケットを送信し、その応答にDNSサーバがUDPパケットを送信する。つまり、２パケットのやり取りだけ行う。このことにより、DNSサーバは正規ユーザにキャッシュを持たせる仕様に比べ、正規ユーザにキャッシュを持たせない本システムの仕様の方が負荷は多少高くなるが、正規ユーザがWebサーバにアクセスする際の応答性は初回アクセスと変わらないため、許容範囲であると考える。

３．攻撃プログラムの解析とruleファイルの作成

IDSサーバでは、DDoS攻撃を検知しなければならないため、攻撃プログラム毎にruleファイルを事前に作成する。そこで、ruleファイルを作成する際、攻撃プログラムの特徴を理解しなければruleファイルを作成することができない。よって、攻撃プログラムがどのような攻撃を行うかをパケットキャプチャして解析し、既に公になっているruleファイルの作成方法を本システムに応用し作成した。

３．１　DDoS攻撃の種類

Webサーバが受けうるDDoS攻撃は大きく二つ種類（脆弱性攻撃、侵入）に分類される。

（1）システム資源への攻撃

プロトコルやWebサーバなどの脆弱性を利用し、Webサーバのシステム上における資源（最大同時接続数、最大使用メモリ容量など）や処理能力を消費させることで、サービスを停止する攻撃のことである。

（2）侵入攻撃

プロトコルやWebサーバなど脆弱性を利用し、システム内部に侵入し、HP改ざんや不正プログラム実行などを行う攻撃のことである。ほかに、正規の権限を持つ者による内部犯行もこれに含まれる。

以上の分類のうち、侵入攻撃の正規の権限を持つ者による内部犯行は、システムの運用上の問題であるため、本研究での検証のための攻撃種類から除外する。また、システム資源への攻撃のうち、実際にサービスを停止するなど、多大な影響を与えうる下記の攻撃プログラムを使用し、検証を行った。

攻撃プログラム

slowloris.pl
killapache.pl
ab（apache bench）

３．２　slowloris攻撃

slowloris攻撃とは、Webサーバを構築するためのソフトウェア（Apache）に対する攻撃手法で、ターゲットとなるApacheとTCPコネクションを行い、接続を開きメッセージヘッダを送信する。その後、時間を空け少しずつメッセージボディを送信し続けることで接続を開いたままの状態を維持する。接続を開いたまま維持したコネクションを多数作り出し、Webサーバの最大同時接続数（MaxClient数）まで接続することでWebサーバが新たに接続する正規ユーザを受け入れることができない状態を作り出す。このことにより、正規ユーザがアクセスすることを困難にさせ、Webサービスを妨害する攻撃である（図４）。

この攻撃を解析するために攻撃ホストでプログラム「slowloris.pl」をWebサーバに向けて実行し、IDSサーバでパケットをキャプチャした。図５のパケットはslowloris攻撃のTCPコネクション部分である。図５よりIP addresss 192.168.136.10（攻撃ホスト）から192.168.3136.140（Webサーバ）の80番portに

図４　slowloris攻撃の概要図

```
16:43:09.432803 IP 192.168.136.10.36371 > 192.168.136.140.http: Flags [P.], seq
0:232, ack 1, win 92, options [nop,nop,TS val 109302582 ecr 92912752], length 232
0x0000:  4500 011c 6b78 4000 4006 3c7c c0a8 880a  E...kx@.@.<|....
0x0010:  c0a8 888c 8e13 0050 dbeb b843 43c5 d758  .......P...CC...X
0x0020:  8018 005c ee3a 0000 0101 080a 0683 d336  ...¥.:.........6
0x0030:  0589 bc70 4745 5420 2f20 4854 5450 2f31  ...pGET./.HTTP/1
0x0040:  2e31 0d0a 486f 7374 3a20 3139 322e 3136  .1..Host:.192.16
0x0050:  382e 3133 362e 3134 300d 0a55 7365 722d  8.136.140..User-
0x0060:  4167 656e 743a 204d 6f7a 696c 6c61 2f34  Agent:.Mozilla/4
0x0070:  2e30 2028 636f 6d70 6174 6962 6c65 3b20  .0.(compatible;.
0x0080:  4d53 4945 2037 2e30 3b20 5769 6e64 6f77  MSIE.7.0;.Window
0x0090:  7320 4e54 2035 2e31 3b20 5472 6964 656e  s.NT.5.1;.Triden
0x00a0:  742f 342e 303b 202e 4e45 5420 434c 5220  t/4.0;..NET.CLR.
0x00b0:  312e 312e 3433 3232 3b20 2e4e 4554 2043  1.1.4322;..NET.C
0x00c0:  4c52 2032 2e30 2e35 3033 6c33 3b20 2e4e  LR.2.0.503l3;..N
0x00d0:  4554 2043 4c52 2033 2e30 2e34 3530 362e  ET.CLR.3.0.4506.
0x00e0:  3231 3532 3b20 2e4e 4554 2043 4c52 2033  2152;..NET.CLR.3
0x00f0:  2e35 2e33 3037 3239 3b20 4d53 4f66 6669  .5.30729;.MSOffi
0x0100:  6365 2031 3229 0d0a 436f 6e74 656e 742d  ce.12)..Content-
0x0110:  4c65 6e67 7468 3a20 3432 0d0a            Length:.42..
```

図５　slowloris攻撃のパケットキャプチャ

```
alert tcp any any -> 192.168.136.140 80 (msg:"slowloris test
rules 192.168.136.140"; flags:P+; dsize:229;
content:"Content-Length: 42"; sid:111222330)
```

図６　slowloris攻撃のruleファイル

送信されていることがわかる。また、Push flagが立っているとこも確認できる。このパケットの全体のsizeは229byteである。さらに、payload部分で「Content-Length: 42」という文字列が確認できる。これらの特徴を元に下記のruleファイルを作成した（図６）。

３．３　apache killer攻撃

apache killer攻撃とは、HTTP Rangeヘッダーに複数の重なり合う範囲を指定し、Webサーバの使用メモリを増大させる攻撃である。HTTP Rangeヘッダーとは、データを分割ダウンロードする際用いられるヘッダーである。例えばWebサーバに1300byteのコンテンツが存在していたとする。このコンテンツを３分割でダウンロードする時、コンテンツの範囲をHTTP Rangeヘッダーに「0-300」「301-800」「801-1300」のように入れて指定しダウンロードする。しかし、RFC2616ではこの範囲を重ねて指定することが可能となっている。よって、「0-300」「0-301」「0-302」と指定できるため、指定する範囲の数が大幅に増加しそれに伴い、生成するパケットの数が増加する。その結果、Webサーバのメモリを大量に消費させて負荷をかけることが可能となる。

この攻撃を解析するために攻撃ホストでプログラム「killapache.pl」をWebサーバに向けて実行し、IDSサーバでパケットをキャプチャした（図７）。IP address 192.168.136.10（攻撃ホスト）から192.168.136.140（Webサーバ）の80番portに送信されていることがわかる。また、window sizeが92byteであり、パケットのpayload部分が1450byte以上であることが確認できる。さらに、Range header部分に「5-」という文字列が複数確認できる。これらの特徴を元にrule

```
00:37:22.339685 IP 192.168.136.10.54987 > 192.168.136.140.http: Flags [.], seq
0:1448, ack 1, win 92, options [nop,nop,TS val 12848932 ecr 6991160], length 1448
0x0000:  4500 05dc 3457 4000 4006 6ed2 c0a8 8814  E...4W@.@.n.....
0x0010:  c0a8 888d d6cb 0050 54a6 3b72 ed7a 8ce2  .......PT.;r.z..
0x0020:  8010 005c 058b 0000 0101 080a 00c4 0f24  ...¥...........$
0x0030:  006a ad38 4845 4144 202f 2048 5454 502f  .j.8HEAD./.HTTP/
0x0040:  312e 310d 0a48 6f73 743a 2061 7061 6368  1.1..Host:.apach
0x0050:  652e 6368 6f63 6f0d 0a52 616e 6765 3a62  e.choco..Range:b
0x0060:  7974 6573 3d30 2d2c 352d 302c 352d 312c  ytes=0-,5-0,5-1,
0x0070:  352d 322c 352d 332c 352d 342c 352d 352c  5-2,5-3,5-4,5-5,
0x0080:  352d 362c 352d 372c 352d 382c 352d 392c  5-6,5-7,5-8,5-9,
0x0090:  352d 3130 2c35 2d31 312c 352d 3132 2c35  5-10,5-11,5-12,5
                    ---中略---
0x0410:  2d31 3239 312c 352d 3132 3932 2c35 2d31  -1291,5-1292,5-1
0x0420:  3239 332c 352d 3132 3934 2c35 2d31 3239  293,5-1294,5-129
0x0430:  352c 352d 3132 3936 2c35 2d31 3239 372c  5,5-1296,5-1297,
0x0440:  352d 3132 3938 2c35 2d31 3239 390d 0a41  5-1298,5-1299..A
0x0450:  6363 6570 742d 456e 636f 6469 6e67 3a20  ccept-Encoding:.
0x0460:  677a 6970 0d0a 436f 6e6e 6563 7469 6f6e  gzip..Connection
0x0470:  3a20 636c 6f73 650d 0a0d 0a              :.close....
```

図７　apache killer攻撃のパケットキャプチャ

```
alert tcp any any -> 192.168.136.141 80 (msg:"slowloris test
rules 192.168.136.141"; flags:P+; dsize:229;
content:"Content-Length: 42"; sid:111222330)
```

図８　apache killer攻撃のruleファイル

ファイルを作成した（図８）。これらの攻撃はapache version2.1.17のRange header DoSの脆弱性に起因する。また、今回の研究で使用しているApacheのversion2.2.22ではこの脆弱性は改善されているため、攻撃が失敗する。よって、検証の際はApacheのversionを2.1.17にdown gradeして検証を行った。しかし、その他の設定はversion 2.2.22のままで行った。

３．４　apache benchによる攻撃

apache benchとは、Apacheに標準で付属しているベンチマークテストツールのことで、HTTP リクエストを大量に送信しレスポンスが返ってくるパケット数、時間などでApacheの性能評価をすることを目的としたツールである。例えば、200件のリクエストを10の同

時接続で20回行い、レスポンスタイムやfailed requestsの回数などで性能を評価する。今回はこのapache benchを用いWebサーバに大量のリクエストを送ることでWebサーバに負荷をかけ攻撃を行った。

この攻撃を解析するために攻撃ホストからプログラム「ab」をWebサーバに向けて実行し、IDSサーバでパケットをキャプチャした（図９）。IP address 192.168.136.10（攻撃ホスト）から192.168.136.140（Webサーバ）の80番portに送信されていることが分かる。また、SYN flagが立っておりwindow sizeが5840byteでpayload部分には何もないことが確認できる。これらの特徴を元にruleファイルを作成した（図10）。３種類の攻撃を解析し、それぞれに対応したruleファイルを作成した。このruleファイルは事前に作成しIDSサーバに置かれ、このファイルとマッチングしたパケットが通過すると、alertを出力する。

図６、８及び10で送信先IP addressを192.168.136.140と記述したが、本システムのWebサーバのIP addressの範囲は192.168.136.140～149であるため、それぞれのIP addressに対応したruleファイルを作成する必要がある。さらに、送信元のIP addressを「any」とすることで、どの攻撃ホストからの攻撃に対しても、これらのruleファイルを適用させている。

```
19:38:47.446282 IP 192.168.136.10.56759 > 192.168.136.140.http: Flags [S],
seq 1598686784, win 5840, options [mss 1460,sackOK,TS val 69590073 ecr
0,nop,wscale 6], length 0
0x0000:  4500 003c da79 4000 4006 ce5a c0a8 880a  E..<.y@.@..Z....
0x0010:  c0a8 888c ddb7 0050 5f4a 0640 0000 0000  .......P_J.@....
0x0020:  a002 16d0 7b58 0000 0204 05b4 0402 080a  ....{X..........
0x0030:  0425 dc39 0000 0000 0103 0306            .%.9........
```

図９　apache benchによる攻撃のパケットキャプチャ

```
alert tcp any any -> 192.168.136.140 80 (msg:"apache bench
rules 192.168.136.140"; flags:S+; window:5840; dsize:0;
sid:112222330)
```

図10　apache benchによる攻撃のruleファイル

４．dual apacheシステム

WebサーバのIP address変更時に、ネットワーク設定ファイルの書き換えとWebサーバ自身のnetworkサービスをrestartする。networkサービスをrestartする際に、LAN内でIP addressが重複しないようARPリクエストを送信し返答を待つ。そのARPリクエストの返答待ち時間があるため、12.49secの長い時間が必要である。また、Webサーバ自身のnetworkサービスをrestartしている期間は、正規ユーザがWebサーバに対してアクセスできないという問題がある。そこで、Webサーバをもう一台用意し２台稼動させる（図11）。このことにより、攻撃を検知し、一台のWebサーバがIP addressを変更するためnetworkをrestartした場合でも、待機状態にあるWebサーバに正規ユーザを誘導することで、Webサーバの切り替えにかかる時間を12.49secから2.087secまで短縮することができる。このWebサーバを２台稼動させるシステムを以後「dual apacheシステム」と呼ぶ。

また、IDSサーバで動かすプログラムでは、Webサーバが一台の場合を想定しているため、

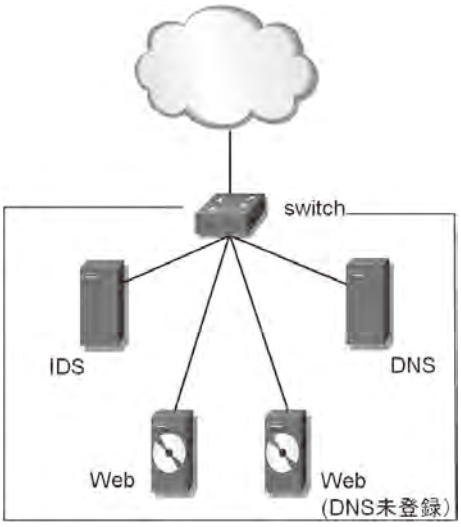

図11　dual apacheシステム構成図

そのままではdual apacheシステムを実現することができない。IDSサーバ内の回避プログラムは、DNSサーバのzoneファイルのAレコードに書かれているIP addressを持っているWebサーバ（以後「誘導中のWebサーバ」と呼称する）と待機状態にあるWebサーバを交互に入れ替え、その際要請をするIP addressが異なるため修正が必要である。例えば、IP addressが192.168.136.140の現在誘導中のWebサーバをA、IP addressが192.168.136.141の待機状態のWebサーバをBとし、140～149の順に誘導する。IDSサーバが攻撃を検知した時、Aのサーバには192.168.136.142のIP addressを申請し、DNSサーバには192.168.136.141のIP addressを申請する。これにより、Bのサーバへの誘導が可能となる。さらに、IDSサーバが攻撃を検知した場合、Bのサーバに192.168.136.143のIP addressを申請し、DNSサーバに192.168.136.142のIP addressを申請する。このように、申請先のIP addressと申請するIP addressが回避システムの場合と異なるため修正を行い、新たにIDSサーバ側プログラムを作成した。

説明上の便宜上WebサーバのIP addressを140～149の順に誘導するように記述したが、本システムでは、このIP addressの範囲のうちランダムに選択し、誘導している。

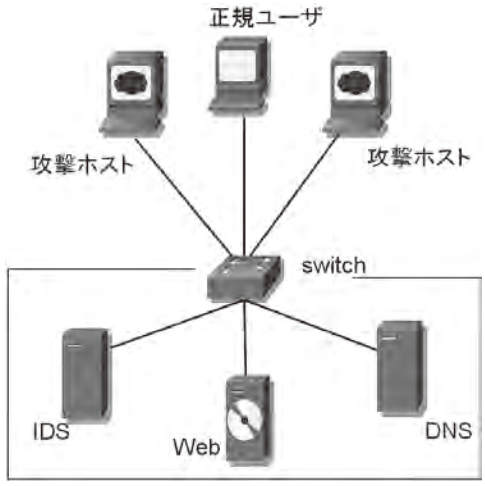

図12　検証システム構成図

5．結果と考察

正規ユーザに対する本システムの可用性を評価するため、攻撃ホストが「slowloris攻撃」と「apache killer攻撃」、「apache benchによるサイトに負荷をかける攻撃」の3種類の攻撃を行い、この間正規ユーザを模して、Webサーバの80番portにtelnet接続しコンテンツを取得後に接続を切り、再度接続して同じことを繰り返す（図12）。この接続で、Webサーバにあるコンテンツの取得が成功した割合で本システムの性能評価を行った。本システムの対策前、対策後、さらにWebサーバを2台稼動させた場合（dual apache システム）を比較した（表1）。本研究の目的は、「IPバージョンに依存しない」と「正規ユーザに影響を与えない」ことであったが、IP addressを利用しているだけであるため、IPv4 やIPv6 などのバージョンによる依存関係はない。しかし、表1より攻撃の種類によっては正規ユーザに影響を与えてしまう結果になった。

表1　コネクション成功率

攻撃手法	対策前	対策後	dual apache
No attack	100.0%	100.0%	100.0%
apache bench	0.0%	99.9%	100.0%
slowloris	0.0%	47.4%	57.8%
apache killer	46.2%	87.4%	93.4%

試行回数1,000回

5．1　slowloris攻撃

slowloris攻撃を対策前のWebサーバに攻撃を行うと正規ユーザはまったくWebサーバのアクセスすることができなくなった（アクセス成功率0％）。しかし、対策後のWebサーバに攻撃を行うと47.4%アクセス成功率が上がった。また、dual apacheシステムに対し攻撃を行うと対策後に比べアクセス成功率が10.4%上昇し57.8%となった。この結果から本システムはslowloris攻撃に対して有効であると考えられる。

しかし、飛躍的に成功率が上がらなかったのは、slowloris攻撃にトレース機能がついていたことが原因と考えられる。トレース機能とは、WebサーバのIP addressを変更しても、再度DNSサーバに正引きアクセスを行い、IP addressを取得し再度攻撃を行う特徴のことである。

また、dual apacheシステムで飛躍的に成功率が上がらなかったもうひとつの原因として、WebサーバのIP addressを変更してから、攻撃を検知して再びWebサーバのIP addressを変更するまでの時間が短すぎることが考えられる。ちなみに、WebサーバのIP addressの変更にかかる時間は平均で12.49secである。この期間に攻撃を２回検知しIP addressを変更するとその間、２つのWebサーバにアクセスが不可能となる。そのため、飛躍的に成功率が上がらなかったと考えられる。

飛躍的に成功率が上がらなかったが、47.4%まで上昇した。これは、攻撃が失敗してからランダム時間空けて再度攻撃するため、Webサーバがサービスを提供できる時間ができ、その間に正規ユーザWebサーバにアクセスできたためアクセス成功率が向上した（図13）。

５．２　apache killer攻撃

apache killer攻撃を対策前のWebサーバに攻撃を行うと正規ユーザは46.2%の割合でアクセスが成功した。対策後のWebサーバへのアクセス成功率は87.4%となり41.2%上昇した。また、dual apacheシステム対策後のWebサーバへのアクセス成功率は93.4%でWebサーバが１つの場合と比べ、6.0%上昇した。

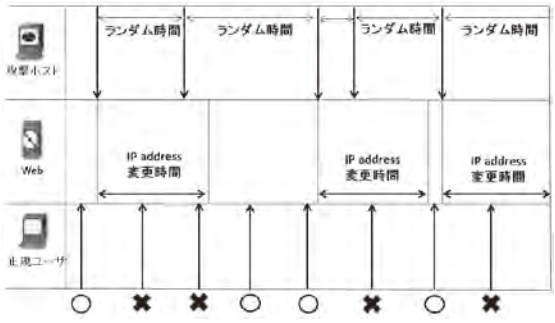

図13　ランダム時間によるアクセス成功図

このアクセス成功率が向上した結果から、本システムはapache killer攻撃に対して有効であると考えられる。

５．３　apache benchによる攻撃

対策前のWebサーバにapache benchによるサイトに負荷をかける攻撃を行うと正規ユーザのアクセス成功率は0.0%で全くアクセスすることができなかった。対策後であると、99.9%まで上昇した。また、dual apacheシステム対策後の正規ユーザのアクセス成功率は100.0%で、必ずアクセスが成功する結果となった。

対策後のアクセス成功率は99.9%という結果については、WebサーバがIP addressを変更するためnetworkをrestartしている間にアクセスしたため失敗してしまったため、100%にならなかったと考えられる。dual apacheシステムは、最初のアクセスを２台目のWebサーバに誘導しアクセスが成功したため、アクセス成功率が100%になったと考えられる。

６．おわりに

本研究では、DDoS攻撃によるWebサーバに対する被害軽減を目的としたDDoS攻撃回避システムを構築し、正規ユーザによるアクセス成功率での性能評価を行った。

apache benchによる攻撃の場合、WebサーバのIP addressを変更した際に新たにDNSサーバに正引きアクセスすることなく攻撃プログラム自体が終了するため、本システムの有効性を示すことができた。しかし、slowloris攻撃やapache killer攻撃は、WebサーバのIP addressを変更しても、これらの攻撃ツールが再度DNSサーバに正引きアクセスを行い、IP addressを取得し再度攻撃を行う特徴がある。この様なトレース機能がある攻撃の場合、本システムの性能が十分に発揮できないと考えられる。

また、このシステムの他の問題点として、攻撃検知のruleが一度の攻撃パケットの通過で検

知してしまうため、具体的被害が発生していない時点でIP addressの変更を行ってしまう。そのため、IP addressの切り替え時間が全体的に増加し、正規ユーザがサーバにアクセスできない時間が増加する問題点がある。

そこで今後改良すべき点は、ある程度攻撃パケットが通過し、被害が出始めたらIP addressを変更するように仕様変更すべきであると考えられる。さらに、IP address切り替えにかかる12.49secの間は切り替えを行わないよう改良すべきであろう。これにより、IP address切り替え時間に正規ユーザがアクセスしてWebサーバへのアクセスが失敗してしまうことがなくなり、トレース機能を持った攻撃に対してもさらに性能を発揮できると考えられる。

今回は検証のために、被攻撃サーバとしてWebサーバを構築したが、IP addressを変更によって回避するため、サーバの種類が違っても本回避システムを適用できると考えられる。

【参考文献】

［1］特集「DoS攻撃」，情報処理，pp. 428-499，Vol. 54 No. 5 May. 2013.
［2］佐竹康宏，大倉一浩，“DDoS攻撃対策におけるフロー識別情報を用いたパケット制御の評価”，信学技報，Vol. 105, No. 527（TM2005-49），pp. 43-48，Jan. 2005.
［3］日下貴義，角将高，馬場達也，稲田勉，“Mobile IPv6 技術を応用した輻輳型DoS攻撃回避方法の提案”，信学技報，ISEC2004-29，pp. 105-112，2004.
［4］snort, http://www.snort.org