6V-03

IDSを用いたDDoS攻撃の検知

山田 洋之† 久保田 光一‡
中央大学大学院 理工学研究科 情報工学専攻†‡

概要: インターネットを通じたサービスは広く普及しているが, そのサービスを妨害する行為も増加している. DDoS 攻撃と呼ばれるサービス妨害が特に増加傾向にあり, 攻撃手法が多岐にわたり, かつ新たな手法も次々と現れるため, 有効な対策を取ることが困難である. 本研究では IDS を使用して NTP を利用したリフレクション DDoS 攻撃を検知することを目的とする.

キーワード: DDoS 攻撃, IDS

1 序論

1.1 背景

近年ではサイバー攻撃のなかでも DoS 攻撃, DDoS 攻撃による事件の増加が問題となっている [1]. 発生件数自体の増加に加えて, 攻撃の停止と引き換えに金銭を要求するなどその事件性はより高くなっている. また DDoS 攻撃を依頼によって実行する業者の出現によって, 今後は企業のサーバだけでなく個人のサーバでも攻撃を受ける可能性がある. そのため個人でも行える DDoS 攻撃への対策が必要である.

1.2 DoS 攻撃

Denial of Service (DoS) 攻撃とはインターネット上で通信量を意図的に増加させることで, 通信の処理を行っているサーバや回線の帯域の処理能力を占有し, システムを使用困難な状態にする, もしくはシステムをダウンさせる攻撃のことである.

1.3 DDoS 攻撃

Distributed Denial of Service (DDoS) 攻撃とは図 1 に示すようにウィルスやセキュリティホールが原因で遠隔操作を受け付けてしまう状態になったボットと呼ばれるコンピュータが一斉に通信量を増加させシステムをダウンさせる攻撃のことである. 数千から数万のコンピュータが同時に攻撃対象へ通信を行うため DoS 攻撃に比べて通信量は大規模になり, 必然的に被害も大きなものとなる. また DoS 攻撃とは異なり, 攻撃を実行しているのは攻撃者のマシンではなく, 遠隔操作されているマシンであり, その所有者は攻撃に無関係である. そのため, 攻撃目的で送られてきたパケットの送信元 IP アドレスを調べても攻撃者を特定できないという問題点がある. 初期の DDoS 攻撃はサービス妨害のみを目的としていたものだったが, 近年では攻撃の停止と引き換えに金銭を要求する脅迫手段として使われる事件や, DDoS 攻撃をサービスとして提供する業者も現れ, 知識を持たない高校生が DDoS 攻撃を依頼し検挙されるといった事件も発生している.

1.4 リフレクション攻撃

DDoS 攻撃は近年, 数百 Mbps を超える大規模な攻撃が発生してる. このような大規模な攻撃はその攻撃手法からリフレクション攻撃と呼ばれる. 通信プロトコルの中でも, データサイズの小さなコマンドに対して, 大きなデータを返す仕様を悪用して行う DDoS 攻撃である. 従来の攻撃と異なり, プロトコルの仕様によってデータサイズが増幅されたパケットが攻撃対象に送られるため, より少ないコンピュータの台数で規模の大きな攻撃を行うことができる. またデータサイズの増幅に利用されたサーバや増幅されたパケットが通過するネットワークに負荷をかけるという問題もある.

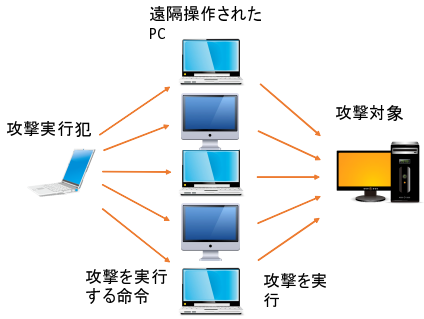

図 1 DDoS 攻撃の様子

1.5 IDS

Intrusion Detection System (IDS) とはコンピュータやネットワークへの不正なアクセスやデータの送受信を検知し通知するシステムのことである. その侵入検知の方法によって不正検出型と異常検出型に分類される. 不正検出型ではあらかじめ登録されたシグネチャと呼ばれるルールファイルを参照して既知の攻撃を検出することができる. 異常検出型ではネットワークのトラフィックやログイン時刻を参照して通常とは異なる振る舞いを検出する. また, その監視場所によってもホスト型とネットワーク型に分類される. ホスト型は保護したいコンピュータにインストールしその 1 台のみを監視するものであり. ネットワーク型は接続しているコンピュータネットワーク内を流れるパケットの収集, 解析を行うものである.

1.6 Snort

Snort は Martin Roesch によって開発されたオープンソースの不正検出型, ネットワーク型の IDS である [2]. 公式コミュニティが活発であり, 既知の不正アクセスや攻撃を対象にした基本的なシグネチャのセットとは別に, 新しい攻撃などを対象にしたシグネチャが公式のコミュニティによって頻繁に更新されているのが特徴である. また, 個人でもシグネチャを制作し Snort に簡単に適用することができるため, それぞれの使用環境に合わせた設定を自由に行うことが可能である.

2 DDoS 攻撃の検知方法

本研究では DDoS 攻撃なかでもリフレクション攻撃に分類される NTP リフレクション攻撃を対象に DDoS 攻撃であるかどうかを判定する基準を統計的に作成する. リフレクション攻撃ではプロトコルや機器の仕様を悪用して攻撃が行われるため, 通常の通信と攻撃を目的とした通信の判別が難しい. そのため事前に攻撃目的で

IDS based detection of DDoS attack

† Hiroyuki YAMADA, Information and System Engineering Course, Graduate School of Science and Engineering, CHUO University
‡ Koichi KUBOTA, Information and System Engineering Course, Graduate School of Science and Engineering, CHUO University

はない正常な NTP による時刻合わせのパケットの送信回数とパケットサイズの平均値を求めておく．具体的には IDS で NTP に関する通信のログを取りこのログを解析した数値を元に平均値との比較を行う．図 2 にログの一部を抜粋したものを示す．このログから 3 行目のパケットが送信された日時と送信元 IP アドレス, 最終行のパケットのサイズである Len の数値を使用する．送信元 IP アドレスごとに単位時間あたりの出現回数と送信してきたパケットサイズを求め, 事前に求めた正常な通信の値とそれぞれを比較する．

```
[**] [1:10000002:0] Type2 NTP access [**]
[Priority:  0]
12/26-15:26:54.541465 133.91.64.24:123 ->
192.168.1.227:123
UDP TTL:253 TOS:0x0 ID:43376 IpLen:20 DgmLen:76
Len:  48
```

図 2　通常の NTP 通信ログの内容

また, 実際の攻撃では NTP サーバにアクセスした IP アドレスの一覧を最大 600 件まで表示する「monlist」という NTP のコマンドが使用される．このコマンドを使用して「ntpdc -nc monlist 133.91.64.24」を実行した際のログを図 3 に示す．このコマンドが使用された攻撃の場合は通常の通信とは異なり, 送信元が未知の IP アドレス (こちらから NTP のパケットを送信したことがない IP アドレス) でポート番号が 123 という特徴が現れる．さらに, 「monlist」コマンドによる通信では通常の時刻合わせを目的とした通信に比べて送信されるパケットの数が多くなる．そこで, この特徴に単位時間あたりの IP アドレスの出現回数とパケットサイズを組み合わせて攻撃であるかどうかの判断を行う．

```
[**] [1:10000002:0] Type2 NTP access [**]
[Priority:  0]
12/26-15:43:47.415385 133.91.64.24:123 ->
192.168.1.227:33963
UDP TTL:253 TOS:0x0 ID:60086 IpLen:20 DgmLen:196
Len:  168
```

図 3　「ntpdc -nc monlist 133.91.64.24」を実行した際のログの内容

3　システム概要

本研究では, オープンソースであり, 手軽に多くの OS で利用できる, 細かな設定変更が可能という点から IDS に Snort を使用する．1.6 節で述べたように Snort はシグネチャを使用する不正検出型の IDS である．監視対象のネットワークを設定するとそのネットワークを対象に通過するパケット内容をチェックし, もしシグネチャで定義されたものであれば管理者に通知を行う．通知方法はコンソールに表示を行う, 図 2 や図 3 のようなログを alert というファイル名で作成するといった動作から選択することができるようになっている．またシグネチャに対応したパケットを検知した時には動作中に通過したすべてのパケットについて記述されたログファイルを作成する．

実験では NTP に関する通信を検知するシグネチャを作成し, alert ファイルを作成することで通知させ, alert ファイルに記述されたログを解析することでその NTP に関する通信が正常な通信か DDoS 攻撃であるかの判断を行う．

3.1　シグネチャ

NTP の通信を調べるために使用するシグネチャを図 4 に示す．ここではシグネチャを 2 個追加する．これらをディレクトリ/etc/snort/rules に配置し Snort の設定ファイルである snort.conf に「include $RULE_PATH type1ntp.rules」「include $RULE_PATH type2ntp.rules」という 2 行を追加するとで Snort がこれらのシグネチャを読み込むようになる [2][3]．$RULE_PATH は snort.conf で定義されている環境変数であり/etc/snort/rules を指定している．自分から時刻を問い合わせる時にパケットを送信する通信を Type1, NTP サーバからパケットが送信されてくる時の通信を Type2 として定義している．DDoS 攻撃では自分が時刻の問い合わせをしていない NTP サーバからパケットが送られてくるため, Type1 で検知される問い合わせ先以外の NTP サーバからの通信は DDoS 攻撃である可能性が高くなる．そのため Type1 と Type2 の両方を検知する事で DDoS 攻撃の判断材料にすることができる．シグネチャの内容を具体的に見ると, 初めの alert はシグネチャの対象となるアクセスを検知した際に警告を通知する命令である．次の udp は対象とするパケットの種類を指定している．次の 2 つの項目は送信元 IP アドレスとポート番号を指定している．ここでは any と指定することですべての IP アドレスやポート番号を指定できる．矢印の後の 2 項目も同様に送信先 IP アドレスとポート番号の指定になっている．ここで指定している$HOME_NET は先ほどの$RULE_PATH と同様に環境変数であり, Snort が動作しているマシンの IP アドレスを登録してある．msg はログに表示するメッセージを記述し, sid はシグネチャの識別 ID である．

```
alert udp $HOME_NET any -> any 123
(msg:‘‘Type1 NTP access’’; sid:10000001)
```

```
alert udp any 123 -> $HOME_NET any
(msg:‘‘Type2 NTP access’’; sid:10000002)
```

図 4　使用するシグネチャを 2 行ずつ異なるファイル type1ntp.rules, type2ntp.rules に格納して登録する

4　実験と結果

シグネチャの追加により, alert ファイルに Type1, Type2 の履歴が格納されるので, それを読み込み, 履歴を解析するプログラムを作成した．仮想マシン上で Snort と ntpd を動作させ, ntp の monlist コマンドを発生させ, 試作プログラムの動作を確認中である．

参考文献

[1] IPA, **“2014 年度セキュリティ事象被害状況調査 -報告書-”**, https://www.ipa.go.jp/files/000043418.pdf, 最終アクセス日 2015 年 12 月 19 日.

[2] **“Snort Users Manual 2.9.7.3”**, https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/086/original/snort_manual.pdf?AWSAccessKeyId=AKIAIXACIED2SPMSC7GA&Expires=1452010548&Signature=dWjLSWwBnIsg8P9pYuCrSLqgpYY%3D, 最終アクセス日 2016 年 1 月 5 日.

[3] **“IDS の導入による不正侵入の検知とネットワーク管理”**, http://www.itmedia.co.jp/help/howto/security/ids/02/05.html#2, 最終アクセス日 2016 年 1 月 5 日.