

TCPセッション管理によるDoS耐性の考察

伊藤大輔[†] 泉裕^{††} 齋藤彰一^{*} 上原哲太郎^{*} 國枝義敏[†]

[†]和歌山大学大学院 システム工学研究科 ^{††}和歌山大学システム情報学センター

^{*}和歌山大学システム工学部

近年のネットワーク管理において、DoS (Denial of Service) 攻撃対策は重要な課題の一つである。DoS 攻撃は、IP アドレスを偽造した TCP パケットを被攻撃者に送信し、過剰処理を行わせることでサービスを妨害する。DoS 攻撃が成功する根本的な原因は、被攻撃者が IP アドレスを元に攻撃者を判断するためである。したがって、IP アドレスによる従来のフィルタリングでは、DoS 攻撃を効果的に抑止できない。

本研究の目的は、偽造パケットによる DoS 攻撃から被攻撃者を守ることである。我々は、TCP 3-WayHandShake において、IP アドレス等を要素として生成した特殊なシーケンス番号による送信者の特定と、セッション確立の際にフィルタリングを行う手法を考案した。本手法では、特定された送信者からのパケットのみ受け取ることで、カーネルへの過剰処理を抑止し、効果的な DoS 攻撃対策を実現できる。

The Research of The TCP Session Control against DoS Attack.

Daisuke Ito[†] Yutaka Izumi^{††}

Shoichi Saito^{*} Tetsutaro Uehara^{*} Yoshitoshi Kunieda[†]

[†]Graduate School of Systems Engineering, Wakayama University

^{††} Center for Information Science, Wakayama University

^{*}Faculty of Systems Engineering, Wakayama University

It becomes rapidly important to protect against DoS Attack in the network management. DoS Attack transmits the TCP packet forged its IP address to the target server(s), then drop its performance because of high overload. It is difficult to avoid DoS attack, because the recent filtering technology almost depend on IP address, and we cannot specify the attacker by IP address forged.

In this paper, we suggest the effective method which purpose is protection against DoS attack using IP address forged. This method works as specifying the source by the sequence number generated by above IP address and so on, then filtering by its sequence number. This paper shows one of the method of effective TCP session control against DoS attack for deterrence of high overload in server(s).

1. はじめに

1.1. 背景

インターネットインフラの拡充に伴い、不正アクセスの件数は急激に増加している。特に近年では、セキュリティ技術レベルの向上が著しいため、「ターゲットに如何に侵入するか」ではなく、「ターゲットに如何にダメージを与えるか」が傾向になりつつある。さらに、送信元 IP アドレスを詐称し、偽造パケットを使用した DoS 攻撃 (2 章にて後述) の回数は年々増加している。カリフォルニア大学の試算によると、世界で発生する DoS(Denial of Services)攻撃の 1 週間当たりの被害は 4000 件に及ぶ[1]。しかし、被害が深刻化しているにも関わらず、DoS 攻撃を防ぐ有効な手段は、現在も確立していない。

一方で、Nimda、CodeRed II に代表されるワームが、自己増殖の過程で DoS 攻撃に類する被害を引き起こす場合もある。しかし、自己増殖の過程で最も被害を受けるのは、サーバよりルータやスイッチ等ネットワーク機器である。ショートパケットの大量発生により、ネットワーク機器の負荷を高める攻撃に対しても、早急な対策が必要である。

DoS 攻撃に限らず、一般的なネットワーク・ホスト防御としてフィルタリング手法がある。しかし、DoS 攻撃は攻撃対象となるサーバやサービスを使用不可能とすることを目的にしているため、送信元の IP アドレスを詐称する場合がある。この結果、フィルタリングによる状況回避はきわめて困難になる。本研究は、従来のフィルタリング手法による DoS 攻撃対策の限界に対し、一つの解決策を提案する。

1.2. 目的

現在、TCP パケットを使用した DoS 攻撃による被害が深刻化していることから、本研究では TCP セッション管理により DoS 攻撃に対する手法の提案と、プロトタイプの実装および検証を行う。ただし、本研究では、ネットワーク上のルータおよびスイッチに対する直接的な解決を目的とせず、DoS 攻撃からサーバ本体を防御するホスト・ベ-

スの手法を提案する。

2. DoS 攻撃の概念

DoS とは、サーバで提供しているサービスを妨害するという攻撃方法である。ここで指す「妨害」とは、対象のホストをハングアップ、フリーズさせることで使用不可能にする、あるいはホストとネットワークへの接続を実質不可能にする行為を意味する。前者の例としては、Out Of Band パケットを送信し、受信した Windows95 を使用不能にする、通称“NUKE”と呼ばれる“OOB 攻撃”等を挙げることができるが、近年問題になっているのは、ネットワークへの接続を断つという後者の妨害方法である。

この攻撃には TCP パケットが使用されることが多い。TCP によるコネクション確立では、3-WayHandShake が用いられる (図 1)。最も代表的

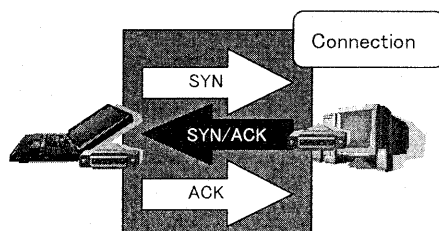


図 1 3-WayHandShake

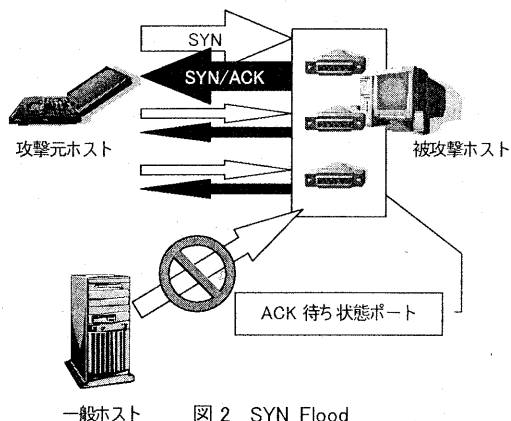


図 2 SYN Flood

な DoS 攻撃は 3-WayHandShake における“SYN Flood”と呼ばれる攻撃である。“SYN Flood”攻撃は、SYN フラグを立てた大量のパケット（以後、SYN パケット）を被攻撃ホスト（攻撃対象のホスト）に送信する。多量の SYN パケットによって被攻撃ホストにおける待機状態のポートを増加させることで、リソースや帯域を不正に消費させる。この結果、他ホストとの通信を困難な状態におくことが可能である（図2）。さらに、SYN Flood 攻撃では、TCP セッションの確立が完了しないので、標準の SYSLOG 等では検出を行うことが困難である。

IDS(侵入検知システム)を導入しているホストやネットワークでは、攻撃の検出が行われ、IP アドレスから攻撃元が判明する可能性がある。しかし近年では、送信元の IP アドレスを偽造した（図3）パケットによる SYN Flood 攻撃が行われることが多い。

SYN Flood 攻撃以外でも、IP アドレスを詐称し、さらにランダムなフラグを立てたパケットを大量送信し、例外処理を行わせるケースも増加している。IP アドレスを詐称している場合、IP アドレスを元にしたフィルタリングは困難である。フィルタリングの要素は、IP アドレスのほかに MAC アドレス、ポートがあるが、DoS 攻撃対策として有効な手法ではないと考える。

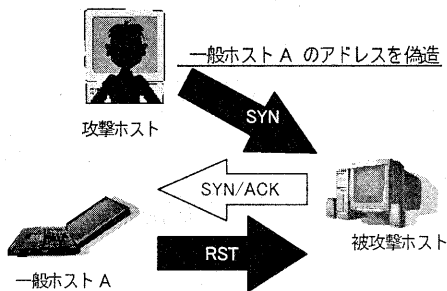


図3 送信元を偽った SYN Flood

3. DoS 対策

通常、不正アクセスと判断できるパケットが検出された場合の対策として、パケットを送信したホストを特定し、同ホストから送信されるパケットの受け取りの拒否を行うフィルタルールを FireWall に記述し、パケットフィルタリングを行うのが一般的である。

しかし、近年の DoS 攻撃の特徴は、前述のように、送信元 IP アドレスを偽造している点にあるため、IP アドレスを元にした不正アクセスホストの特定は事実上不可能である。

偽造パケットを使用した DoS 攻撃への対抗技術としては、ルータなどの中継機器で偽造パケットの送信元を発見する iTrace (ICMP Trace Back) [2] などの手段が検討されているが、すべてのルータにプロトコルが実装されなければ本来の目的を果たすことができないため、現在の段階では有効な手段であるとは言えない。本研究では、SYN Flood 攻撃を代表とする、IP アドレスを詐称したパケット送信に対して、サーバやサービス・プログラムを保護する SPP (SYN Packet Predator) を提案し、プロトタイプの実装および検証を行ったので報告する。

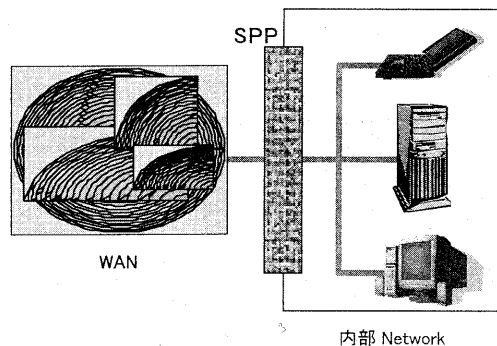


図4 SPP イメージ

4. SPP による Filtering

4.1. 概要

現在使用されているフィルタリングの概念は、パケットの流入/流出時において、フィルタルールに該当するパケットのみを排除し、これ以外は通過させることである[3]。しかし、IP アドレスを元にルールを作成する場合は多いため、IP アドレスの詐称を行ったパケットに対しての強度は決して高くない。

本研究では、流入するパケットを、基本的にすべて排除し、IP アドレス詐称の疑いがないパケットだけを通過させる、という方針に基づいたフィルタリングシステムを提案し、その方針に基づき SPP を作成した。構成図を図 4 に示す。

4.2. SPP の概要

送信された TCP パケットに SYN フラグが立っていた場合、SYN Flood であるか、正規の接続要求であるかは、TCP セッションの確立如何で判明する。TCP セッションの確立完了は、通信を望む送信元が確実に存在し、通信の意思を有するという意

味する。SYN Flood の場合にはセッションは確立しない。

セッションの確立が確定するのは、3-WayHandShake において、ACK フラグの立ったパケット（以後、ACK パケット）が送信された場合である（図 5）（以後、セッション構築の際の ACK パケットを“AU_ACK”とする）。すなわち、結果的に、3-WayHandShake の監視を行うことで、不正パケットの検出は可能である。

通常は、各ホストのカーネルが 3-WayHandShake を監視するため、攻撃であった場合には負荷がかかり、DoS 攻撃が成立する。本案では、SPP が監視を行うことで、内部ホストへの負荷を軽減する。なお、内部ホストとは、SPP に保護される内部ネットワーク内のホストであり、外部ホストとは、内部ホストに対して通信要求を送信する WAN（外部ネットワーク）上のホストである。

外部ホストから内部ホストへの接続要求があった場合、SPP は内部ホストの IP アドレスを使用し、外部ホストとの間にセッションの構築を試みる。セッション構築が成功しなかった場合は、偽造 SYN パケットであったと判断して排除する。成功した場合、外部ホストから見れば内部ホストとセッション構築が行われたように見える。外部ホストとのセッション構築が成功すると、SPP は次に、外部ホストの IP アドレスを使用し、内部ホストとの間にセッションを構築する。この場合も、内部ホス

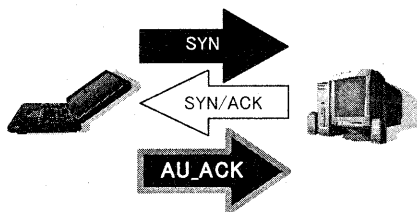


図 5 通常の 3-WayHandShake

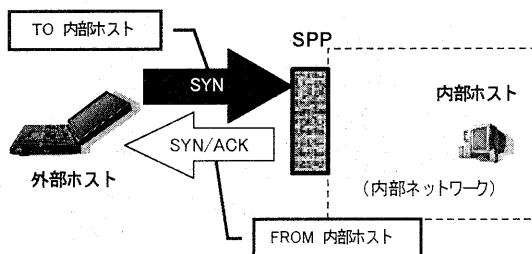


図 6 SYN パケット受信時

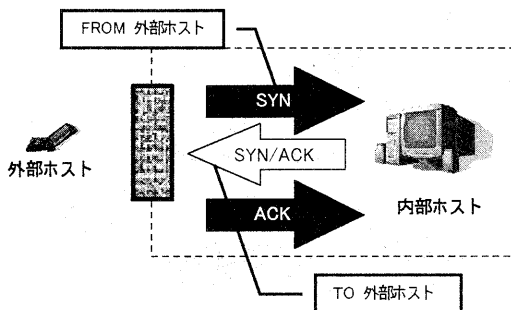


図 7 内部ネットワークにおけるセッションの確立

トから見ると外部ホストとの間にセッション構築が行われたように見える。後の通信は SPP を介して通常通りに行われる。

すなわち、通信路に SPP を介することで、内部ホストから見ると、IP 詐称を行っていない、信用できる (SPP との間に 3-WayHandShake を確立した)

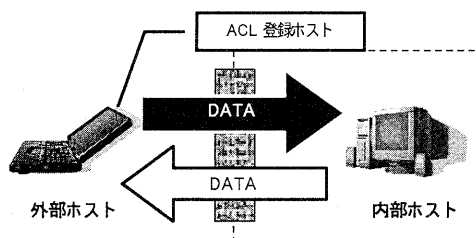


図 8 セッションの確立後のパケット送受信

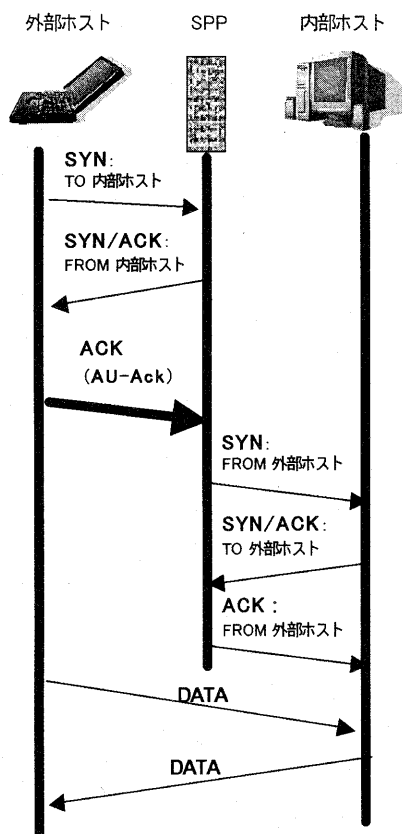


図 9 セッション確立のタイムチャート

外部ホストとのみ、セッションを構築し、通信を行うことが可能となる。

SPP は、TCP パケットのフラグによって、異なる動作を行う。各パケットによる動作を以下に示す。

4.2.1. SYN パケット受信時

外部ホストから内部ホスト宛に送られてきたパケットが SYN パケットであった場合、SPP はパケットを内部ネットワークへは通過させない。一方で、内部ホストを送信元 IP アドレスとした、SYN/ACK フラグが立ったパケット（以後、SYN/ACK パケット）を生成し、外部ホストへ送信する（図 6）。これにより、外部ホストでは、内部ホストが接続要求を受け付けたように見える。

もし送信されたパケットが、SYN Flood による偽造パケットであった場合でも、攻撃対象である内部ホストには、パケットが到達していないため、必要以上に負荷がかかることはない。

4.2.2. ACK パケット受信時

SPP が送信した SYN/ACK パケットに対して、外部ホストが AU_ACK パケット送信した場合、SPP は、内部ホストに対し、外部ホストを送信元 IP アドレスとしたパケットを生成し、今度は外部ホストであるかのように振る舞い、内部ホストと TCP セッションの確立を試みる（図 7）。

SPP と内部ホストとの間でセッションの確立が成功した場合、対象の外部ホストを Access Control List（以後“ACL”）に追加する。ACL は SPP 内に存在するリストで、内部ホストとのセッション構築を許可した外部ホストのデータがリストされている。

このとき、ACL には

- ・ 送信元 IP アドレスとポート番号
- ・ 受信先 IP アドレスとポート番号

などのパラメータの登録を行う。

以後は SPP、ACL に該当するホストから送信されたパケットのみを通過させる。

4.2.3. 他のパケット

パケットが IP アドレスなど ACL の内容と完全に一致するものであれば、内部への通行を許可する (図 8)。

しかし、4.1.1、4.1.2 での手続きを取らなかった ACK パケットや、ほかのフラグの立ったパケットなど、ACL に該当するリストが無いパケットは、SPP によって排除される。

以上の方式において、TCP セッションを確立するためのタイムチャートを図 9 に示す。

4.3. 特定シーケンス番号方式

TCP セッション構築の際、各ホストのセッション構築状態情報はサーバのカーネルメモリにおいてリストされるのが一般的である (この方式を以後、リスト方式とする)。しかし、IP アドレスを詐称した多量の SYN パケットが送信された場合、SPP を起動しているホスト (ゲートウェイ) において、

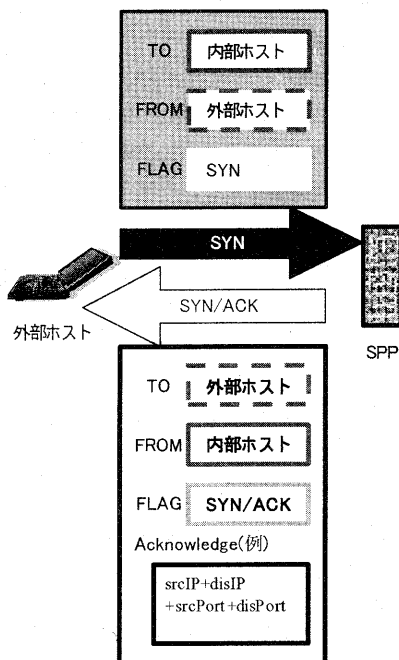


図 10 SYN/ACK パケットの作成

リソース (主にメモリ) が枯渇し、DoS が成立する危険性が生じる。したがって、メモリ枯渇への対策として、特定シーケンス番号方式を考案した。

SPP が SYN パケットを受信した場合、カーネルで送信元をリストせず、送信元 IP アドレス・ポート番号と送信先 IP アドレス・ポート番号を入れ替え、フラグを SYN から SYN/ACK へと単純に変更を行ったパケットを返信する (図 10)。

このとき、IP アドレス、送信ポート番号、受信ポート番号などから作成した特殊な数値を Acknowledge に格納する。Acknowledge は図 11 のように、相手ホストが次に送信してくるシーケンス番号を指定するためのパラメータである。

次に、SPP が AU_ACK パケットを受信した際、ホスト情報がカーネルメモリにリストされていないとしても、ACK パケットの各種パラメータ (送受信双方の IP アドレス・ポート番号など) から再計算し、送信されてきたシーケンス番号と比較する。比較の結果、対象の ACK パケットが AU_ACK であるかどうかを判断することが可能である。

特定シーケンス番号方式は、メモリの浪費だけではなく、偽造された ACK パケットの検出・排除にも効果的である。

実装

SPP は、ゲートウェイなど内部ネットワークと外部ネットワークの境界線に設置することを前提としているが、今回は検証用として、特定シーケンス番号方式を導入した、ホストベースの SPP を作成し、FreeBSD 上で実装を行った。

ここでは、図 12 のように、カーネルを SPP における内部ホストと見立て、カーネルと SPP とで内部セッションを確立させる。具体的にはカーネル内部の IP スレッドの入力関数と出力関数に SPP エンジンを設置して処理を行っている (図 13)。

今回は、ホストベース SPP により、SPP 自体の実現性・有効性を検証する。

評価

今回は synk4.c[4]を使用し、実際に SYN Flood を行って実験した。その結果、カーネルへは偽造した SYN パケットが到達しないことが確認できた。その他、送信元 IP アドレス詐称を行った ACK パケットにおいても同様の結果を得られた。

しかし、CPU やメモリの消費、導入機によるオーバーヘッドなどについては現在評価中である。

最後に

今回は

- ・ SPP
- ・ 特定シーケンス番号方式

の提案を行った。

これらは導入機器の演算能力によって負荷が左右されるという点がボトルネックである。

特に、特定シーケンス番号方式は SYN Flood への高い耐性を持つが、アドレス詐称を行った偽造 ACK パケットを多量に送信された場合、シーケンス番号の再計算に大きな負荷がかかってしまう。

シーケンス番号生成の過程において、生成に簡単なアルゴリズムを用いた場合には負荷の軽減と速さにおいて優れるが、セキュリティ強度が低いものとなる。逆に、複雑なアルゴリズムで生成を行

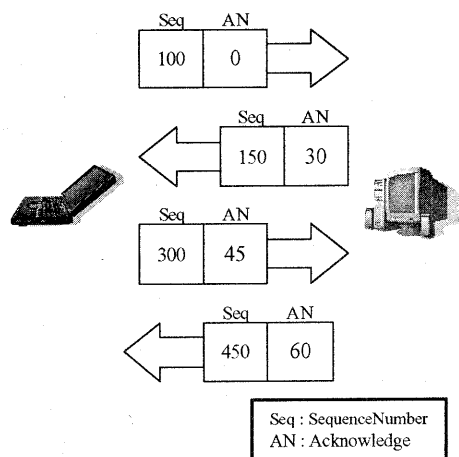


図 11 SequenceNumberとAcknowledge

った場合には、セキュリティ強度的には優れるが、高い負荷がかかる可能性があるため、アルゴリズム採用に関してのトレードオフ決定に難点がある。

一方、特定シーケンス番号方式で生成されるものと偶然同じシーケンス番号で、偽造 ACK パケットが送られてきた場合、内部ホストとの間に接続を誤って確立してしまう可能性も否定できない。

以上のような問題を解決するため、ほかに TTL などの要素を使用した場合の可能性について、現在検討中している。

今後は、SPP の詳細な使用を充実させ、実装、運用した上で、リスト方式と特定シーケンス番号方

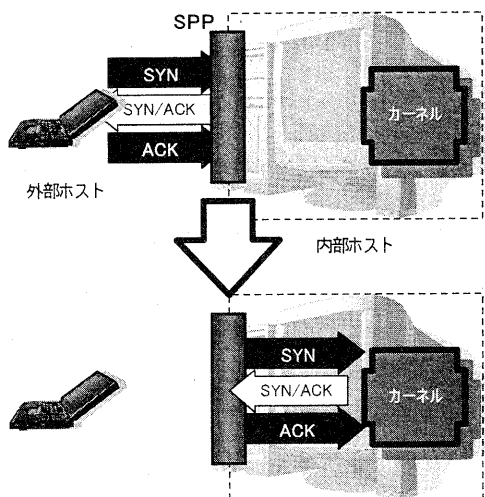


図 12 SPP のイメージ

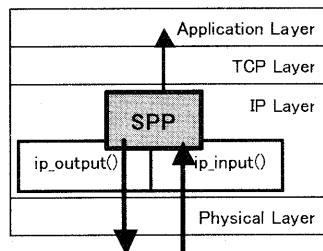


図 13 SPP の内部イメージ

式のオーバーヘッド等の比較, 検討を行っていく.

参考文献

- [1] [http://www.caida.org
/outreach/papers/backscatter/](http://www.caida.org/outreach/papers/backscatter/)
- [2] [http://www.ietf.org
/html.charters/itrace-charter.html](http://www.ietf.org/html.charters/itrace-charter.html)
- [3] Philip Miller , Mastering TCP/IP ,
Olmsha (1998)
- [4] [http://packetstormsecurity.org/
Exploit_Code_Archive/synk4.c](http://packetstormsecurity.org/Exploit_Code_Archive/synk4.c)