社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－DPS－130 (65)
2007－CSEC－ 36 (65)
2007／3／2

throttling による攻撃抑制の効果について

吉田 和幸† 南 浩一‡

†大分大学総合情報処理センター
‡大分大学工学部

あらまし セキュリティホールが残っている古いソフトウェアの存在等を探す scan 攻撃が後を絶たない．大分大学ではこれらの対策として，scan 攻撃である TCP コネクションに対する応答を選択的に遅くし，事実上 scan 攻撃を抑止するシステムを提案し，運用している．本システムは透過型ブリッジとして動作し，ネットワーク境界付近に設置することを想定している．本論文では，運用のログから，本システムによる攻撃抑制の効果について述べる．

キーワード ネットワークセキュリティ，スキャン攻撃，IDS，IDP

The Attack Control System using Throttling

Kazuyuki Yoshida† and Kouichi Minami‡

†Information Processing Center, Oita University
‡Department of Computer Science and Intelligent Systems, Oita University

Abstract There are a lot of scan attacks which look for existence of the old software in which the security hole remains etc. As these attacks measure ,we propose and operate the system which adds delay to TCP connections for scan attacks and deters scan attacks as a matter of fact in Oita University. It implements as a transparent bridge, so that the system can be installed anywhere in a network. In this paper, we describe effect of attack control by this system from the log.

Keyword network security, scan attack, IDS, IDP

1 研究背景

セキュリティホールが残っているコンピュータや特定のサービスを動作させているコンピュータの存在等を探す scan 攻撃が後を絶たない．大分大学で運用している不正侵入検知装置でも日々多くの警告が通知されている[1]．不正アクセスによる侵入を許すと，侵入されたコンピュータが，他のコンピュータを攻撃するための踏み台や，spam の中継，フィッシング詐欺などに利用され，他のユーザやネットワークに被害を及ぼすケースもある．そのためネットワークの管理者は，こまめにログを監視し，適切なフィルタリングを行うなどの対策を行う必要がある．しかし，大学のような研究室単位で多くのコンピュータが管理されている環境では，全てのコンピュータに対策を行なうのは難しい．そこで本稿では，throttling を用いてネットワーク単位で scan 攻

撃を抑制するシステムの提案を行う[2]．本システムは設置場所を選ばない透過型ブリッジとしてネットワークの境界で動作する．(図 1).

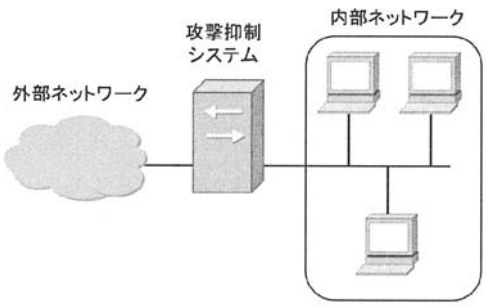

図1 透過型ブリッジによる配置

2 scan 攻撃

2．1 scan 攻撃の定義

本システムは，TCP [3] による scan 攻撃を扱う．TCP コネクション要求（TCP オプションの SYN フラグのみが 1 であるパケット）が外部ネットワークの同一送信元アドレスから複数回送られたとき, scan 攻撃が行なわれたと判断する．

2．2 scan 攻撃の調査

scan 攻撃の特徴を調査するため, 1 週間，大分大学の 1 つのサブネットに送られてくるパケットのログを取った．このネットワークではフィルタリングを行わず,全てのパケットを通過させた．このネットワーク内ではコンピュータは稼動していない．

2．3 scan 攻撃の特徴

上記のログを分析した結果,このネットワークに対して scan 攻撃を行った IP アドレスの数は 23,376 個であった．これらの送信元 IP が,コネクション要求の再転送を何回行っているか調査した．表 1 は送信元 IP アドレスが，1 つのホストに対してコネクション要求の再転送を最大何回行ったかまとめたものである．この表より，scan 攻撃の約 70％が 3 回までしか再転送を実行していないことがわかる．これより，scan 攻撃はコネクションの確立に失敗したとき，早めに諦めていることが予想できる．

1つのホストへの コネクション要求の再転送回数	IP数
1回	930
2回	1185
3回	15585
4回	373
5回以上	5303
計	23376

表1 コネクション要求の再転送回数

3 throttling アルゴリズム

3．1 throttling

本システムでは scan 攻撃を抑制するために，spam 対策などで用いられている throttling の技術を使用した．throttling とは，アクセスに対して一定時間の遅延を挟むことで，意図的に応答を遅らせる手法である．TCP を用いた scan 攻撃に対して throttling を行うことで相手への返答を遅らせる（図 2）．また遅延をかけている間は，遅延をかけているアドレスからのパケットの流量を限定する（図 3）．遅延をかけている間に攻撃者が TCP コネクションを破棄すれば，事実上 scan 攻撃を抑制したこととなる．また TCP コネクションを諦めなくとも，scan 攻撃終了までにかかる時間が長くなるため，攻撃者は多くのホストにアクセスすることが難しくなる．

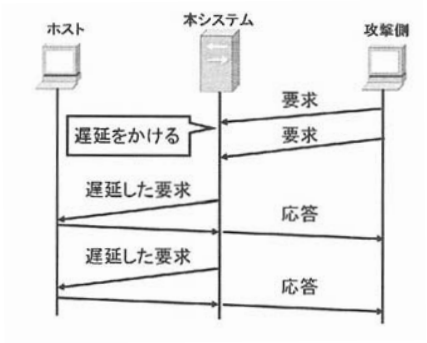

図２　攻撃に対する遅延

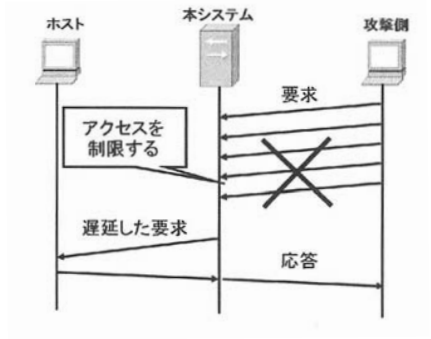

図３　パケットの制限

３．２　考慮点

本システムは複数回のコネクション要求が行なわれたとき scan 攻撃だと判断する．これをそのまま実装すると，全ての TCP に throttling が行なわれることになる．scan 攻撃でないアクセスに影響がでるのは問題があるため，コネクション要求の回数に応じて遅延の量を変化させる．システムは送信元 IP アドレス毎の保持情報としてコネクション要求回数を保持しておく．これを元に，要求回数が少ないときには小さな遅延をかけ，回数が増えるにつれて大きな遅延をかける．ただしこの情報をいつまでも保持しているとscan 攻撃でなくても大きな遅延がかかる可能性がある．そのため，保持情報に最終アクセス時刻を追加する．そして定期的に，保持している全ての IP アドレスの最終アクセス時刻の検査を行い，一定時間アクセスがない IP アドレスの情報を破棄する．

３．３　ホワイトリスト

外部ネットワークの IP アドレスの中にも，遅延をかけたくない IP アドレスが存在する可能性がある．また HTTP 等の，多くのコネクション要求が必要なプロトコルに大きな遅延がかかる可能性もある．これらの IP アドレスやポート番号は，ホワイトリストを用いてあらかじめ除外する．

３．４　throttling アルゴリズム

本システムは，scan 攻撃に遅延をかけるために，全てのパケットに throttling アルゴリズムを適用する．throttling アルゴリズムの手順を図４に示し以下で詳しく説明する．

throttling アルゴリズム：

入力：パケットのデータ，現在時刻

出力：遅延時間

手順１：｛TCP ヘッダの解析｝

パケットが TCP を使用しているか調べる．使用してなければ遅延時間を 0 秒として終了．

手順２：｛ホワイトリストの確認｝

送信元 IP アドレスと TCP ヘッダの宛先ポート番号を調べる．ホワイトリストに該当するものがあれば遅延時間を 0 秒として終了．

手順３：｛SYN フラグの検査｝

TCP オプションを調べる．SYN フラグのみが１であれば手順４へ．そうでない場合は手順５へ．

手順４：｛情報の追加・更新｝

送信元 IP アドレスのコネクション要求回数を 1 増やす．送信元 IP アドレスの最終アクセス時刻に現在時刻を設定する．

手順５：｛遅延時間の算出｝

パケットの送信元IPアドレスのコネクション要求回数を調べ，遅延時間を算出する．

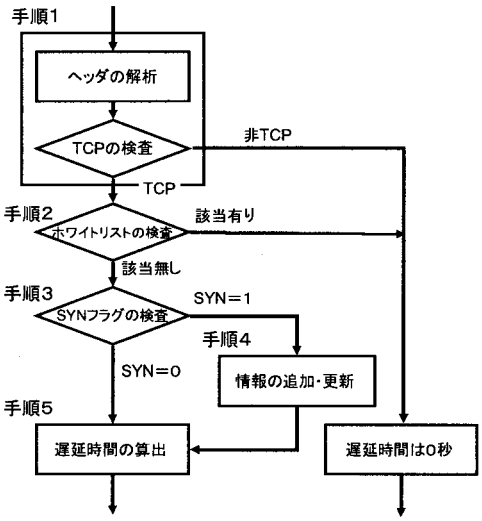

図4　throttlingアルゴリズムの流れ

4　実装

4．1　全体の構成

システムは外部ネットワークから来たパケットをすべて受信し，throttlingアルゴリズムを適用する．その後，内部ネットワークに送信する．内部ネットワークから来たパケットは，すべて無条件に外部ネットワークに送信する．システムの内部構成を以下に示す（図5）．システムは，パケットの受信を行う受信部，パケットにthrottlingアルゴリズムを適用する解析部，パケットに遅延をかけ送信する送信部から構成されている．各部について順に説明する．

4．2　受信部

受信部はネットワークデバイスからパケットの受信を行う．受信したパケットは解析部に送る．

4．3　解析部

解析部は受信部から送られたパケットを受け取り，パケットにthrottlingアルゴリズムを適用する．そのために，送信元IPアドレス毎の保持情報としてコネクション要求回数と最終アクセス時刻を保持する．これらの情報はradish treeに格納し，効率よく検索を行う[4]．パケットにthrottlingアルゴリズムを適用した結果，得られた遅延時間が0秒でなければ，現在時刻に遅延時間を加算して送信時刻を求める．そして得られた送信時刻をパケットに付加する．throttlingアルゴリズムを適用後，パケットを送信部に送る．

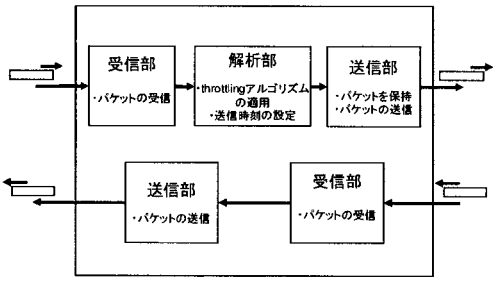

図5　システムの内部構成

4．4　送信部

送信部は，受信部もしくは解析部から送られたパケットを管理し，適切な時間に送信する．そのために送信部は遅延なしキューと遅延ヒープを保持している．遅延なしキューには，送信時刻が付加されていないパケットと内部ネットワークから来たパケットを挿入する．遅延なしキューに挿入されているパケットは直ぐに送信する．遅延ヒープには送信時刻が付加されたパケットを挿入する．遅延ヒープには1つのアドレスに対する最大保持数が定められている．最大保持数以上のパケットは挿入を行わない．挿入できなかったパケットは破棄する．遅延ヒープ内では送信時刻を優先度として送信時刻の早い順に並べている．システムは遅延ヒープの先頭を定期的に調べ，送信時刻に到達しているパ

ケットを送信する.

5　運用評価

5．1　運用環境

システムを図6のように学内LANに設置して運用した．システムは202.253.*.0/24と133.37.*.128/27のネットワークを防御対象とする．202.253.*.1と202.253.*.2のアドレスではSSHサーバを運用している．133.37.*.128/27のネットワークではメールサーバを運用している．

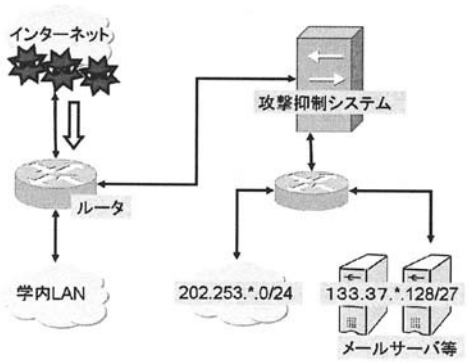

図6　実験環境

5．2　設定

5．2．1　遅延時間の計算

本実験では，throttlingアルゴリズムの手順5で行なう遅延時間の計算を以下のように設定した．ただし最大値を約188秒としている．

遅延時間　＝　$2^{(\beta-1)}$ [1.0×10^{-7}秒]
（β＝コネクション要求回数）

5．2．2　ホワイトリストの設定

学内LANと通信することを考慮して，IPアドレスのホワイトリストには133.37.0.0/16を設定した．また，メールサーバを運用していることから，ポート番号のホワイトリストには，25番，53番，80番，110番，113番，443番を設定した．

5．2．3　最終アクセス時刻の検査間隔

全IPの最終アクセス時刻の検査を30分間隔で行い，30分以上アクセスのないアドレスの情報は破棄するように設定した．

5．3　運用結果

運用結果のログを図7，8に示す．図中の<Add>が記録されている行ではシステムに送信元IPアドレスの保持情報が追加されている．<remove>が記録されている行ではシステムから送信元IPアドレスの保持情報が破棄されている．<drop>が記録されている行では，図3のような流量の制限によりパケットが破棄されている．countはコネクション要求の回数である．システムは，wait=1に対して1.0×10^{-7}秒の遅延をかけている．

図7では，ネットワーク内のSSHサーバを探すscan攻撃を行った後に，scan攻撃に対して反応を返したサーバに攻撃を試みている跡を確認できる．この攻撃は，scan攻撃の途中で遅延量が最大に達し，抑制に成功している．

図8では，ネットワーク内の２９６７番ポートに対して，最終アクセス時刻の検査間隔より長い間隔で，コネクション要求を送信している攻撃の跡が見られる．この攻撃には，小さな遅延しかかからず抑制には到っていない．

6　まとめ

throttlingを用いてネットワーク単位で攻撃を抑制するシステムを開発し，運用を行った．短い間隔でコネクション要求を送信する攻撃に対しては，これを検出し，抑制することができた．しかしながら，長い間隔を空けてコネクション要求を送信するscan攻撃にはうまく動作しない．アドレスの保持期間を長くすると，通常のアクセスにも影響が出る可能性がある．長い間隔を空けてコネクション要求を送信するscan攻

撃への対策方法は，コネクション要求回数以外の方法で行う必要がある．また，どの程度遅延をかけるかについても検討する必要がある．ブラックリストを用いたPAM遅延モジュールによるSSHへの攻撃抑制[5]では，10秒程度の遅延をかけることでSSHへの攻撃を断念させている．

今後は，様々な設定で運用を継続すると同時に，課題である，長い間隔を空けてコネクション要求を送信するscan攻撃への対策を検討していく．

参考文献

[1] 三鑰達真,大野泰宏,吉田和幸: 攻撃の規則性認識を支援する攻撃量時系列変化比較対照表示システム，情報処理学会分散システム/インターネット運用技術シンポジウム2006論文集, pp.55-60, 2006.11

[2] 吉田,南：throttlingを利用したscan攻撃抑制システム，分散／インターネット運用技術シンポジウム2006年度論文集

[3] J.Postel："Transmission Control Protocol", RFC 793, Sep 1981.

[4] Kazuhiko Yamamoto, Akira Kato and Akira Watanabe："Radish – A Simple Routing Table Structure for CIDR", WIDE, 1995 http://citeseer.ist.psu.edu/yamamoto95radish.html

[5] 鈴木,湯浅：ブラックリストを用いたPAM遅延モジュールによるSSHへの攻撃抑制,情報処理学会研究報告(2006-DSM-40), pp.1-5, Mar.2006


```
22:23:51: <add > 210.83.208.155 48024 ==> 202.253.*.0 22 tcpopt=2 count = 1 wait=1
22:23:51: 210.83.208.155 48025 ==> 202.253.*.1 22 tcpopt=2 count = 2 wait=2
22:23:51: 210.83.208.155 48026 ==> 202.253.*.2 22 tcpopt=2 count = 3 wait=4
22:23:51: 210.83.208.155 48027 ==> 202.253.*.3 22 tcpopt=2 count = 4 wait=8
22:23:51: 210.83.208.155 48028 ==> 202.253.*.4 22 tcpopt=2 count = 5 wait=16
22:23:51: 210.83.208.155 48029 ==> 202.253.*.5 22 tcpopt=2 count = 6 wait=32
22:23:52: 210.83.208.155 48030 ==> 202.253.*.6 22 tcpopt=2 count = 7 wait=64
(中略)
22:24:50: <drop> 210.83.208.155 48359 ==> 202.253.*.253 22 tcpopt=2
22:24:50: <drop> 210.83.208.155 48358 ==> 202.253.*.252 22 tcpopt=2
22:24:50: <drop> 210.83.208.155 48356 ==> 202.253.*.250 22 tcpopt=2
22:24:50: <drop> 210.83.208.155 48360 ==> 202.253.*.254 22 tcpopt=2
22:32:45: 210.83.208.155 43388 ==> 202.253.*.1 22 tcpopt=2 count = 64 wait=1879048192
22:32:45: 210.83.208.155 43389 ==> 202.253.*.2 22 tcpopt=2 count = 65 wait=1879048192
22:32:48: 210.83.208.155 43388 ==> 202.253.*.1 22 tcpopt=2 count = 66 wait=1879048192
22:32:48: 210.83.208.155 43389 ==> 202.253.*.2 22 tcpopt=2 count = 67 wait=1879048192
22:32:54: 210.83.208.155 43388 ==> 202.253.*.1 22 tcpopt=2 count = 68 wait=1879048192
22:32:54: 210.83.208.155 43389 ==> 202.253.*.2 22 tcpopt=2 count = 69 wait=1879048192
00:08:14: <remove> 210.83.208.155
```

図7　**SSH**への**scan**攻撃

```
16:29:39 <Add> 202.137.164.250 3813 ==> 202.253.*.192 2967 tcpopt=2 count=1 wait=1
16:29:42 202.137.164.250 3813 ==> 202.253.*.192 2967 tcpopt=2 count=2 wait=2
16:32:15 202.137.164.250 1340 ==> 202.253.*.58 2967 tcpopt=2 count=3 wait=6
(中略)
17:23:08 202.137.164.250 4641 ==> 202.253.*.96 2967 tcpopt=2 count=7 wait=64
18:23:48 <remove> 202.137.164.250
18:47:51 <Add> 202.137.164.250 1291 ==> 202.253.*.0 2967 tcpopt=2 count=1 wait=1
19:46:46 202.137.164.250 4349 ==> 202.253.*.131 2967 tcpopt=2 count=2 wait=2
(中略)
20:44:18 202.137.164.250 1319 ==> 202.253.*.144 2967 tcpopt=2 count = 9 wait=256
21:50:26 <remove> 202.137.164.250
```

図8　ポート**2967**への**scan**攻撃