throttlingを利用したscan攻撃抑制システム

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．アカウント名/パスワードを探すパスワードクラッキングも同様である．これらの攻撃は，1つの IP アドレスから来ることが多い．scan 攻撃である TCP connection 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について提案する．本システムは，透過型ブリッジとして実現している．そのため設置場所を選ばない．
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 Scan 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. There are also many password cracking attacks which look for an account name / password for some server. These attacks may come from a host with one IP address. We propose the system which adds delay to TCP connections for scan attacks and deters scan attacks as a matter of fact. It implements as a transparent bridge, so that the system can be installed anywhere in a network.
Keyword network security, scan attack, IDS, IDP

1. はじめに

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攻撃や，TCPの22番ポートや135番ポート等の特定のポート狙った攻撃が後を絶たない．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，コンピュータの管理者は，こまめにログを監視し，攻撃元の IP に対してフィルタリングを行うなど攻撃の対策を行う必要がある．しかし，大学のように管理者が複数存在し，研究室単位で多くのコンピュータが運用されている環境では，対策が行われないままのコンピュータが存在することや，早めの対策が行われないことがある．そこで本稿では，ネットワーク単位で scan 攻撃や不正

アクセスを抑制するシステムの提案を行う．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動作するため，ネットワークの境界に設置することができる(図 1)．

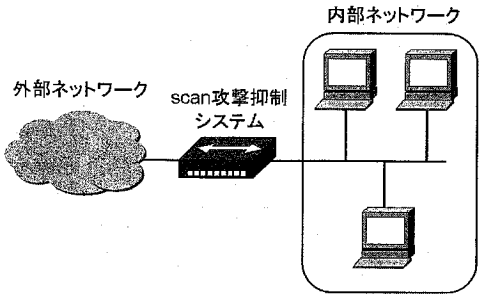

図1 透過型ブリッジによる配置

2. scan 攻撃の抑止方法

ネットワークやコンピュータに不正アクセスが行われる前には scan 攻撃が行われることが多い．scan 攻撃を許すと，攻撃者にネットワークやコンピュータの情報を与えることになる．攻撃者に情報を与えることは精度の高い攻撃につながることになるため，scan 攻撃を防ぎ情報を提供しないことが不正アクセス防ぐ上で重要である．また，早い段階で scan 攻撃を検知することにより，攻撃が行われる前に対策を行うことが可能となる．

本システムは，scan 攻撃の際に使用される TCP コネクションを検知し，TCP コネクションへの応答を選択的に遅くする throttling を用いて scan 攻撃を抑制する．throttling を用いることにより，次の効果を期待できる．

(1)scan 攻撃自体を断念させる

(2)攻撃終了までの時間を長くする

遅延をかけることにより，攻撃者が scan 攻撃を諦めることを目的とする．SSH によるパスワードクラッキング攻撃において，遅延をかけることで攻撃を断念されることができている[1]．scan 攻撃に対しても攻撃抑止の効果を期待できる．scan 攻撃終了までの時間を長くすることで，その間に対策を行うことができる．

既存の scan 攻撃の対策方法には，パケットフィルタリングや IDS，IDP などがある．本システムは，リアルタイムで攻撃に対処するため IDP と似たような部分があるが，攻撃に対して通信をブロックするのではなく反応を遅らせるという手法を用いることに違いがある．

3. throttling のアルゴリズム

3.1 概要

scan 攻撃はツールを用いて自動的に実行されることが多く，大抵は1つの IP アドレスから行われている．また，多くの TCP コネクションを短時間で試行する．そこで，同じ IP アドレスからの TCP コネクションの開始を検知するたびに，その送信元 IP アドレスに対して遅延を増加させていく．つまり TCP コネクション試行回数に応じて遅延をかけていく．これにより TCP コネクションを使用する scan 攻撃に対して throttling を行う．

3.2 考慮点

TCP コネクションの試行回数から遅延時間を決定するときに次の 2 点について考慮する．

3.2.1 正規の利用者

ssh で、パスワードの入力を間違えるなど、正規の利用者も TCP コネクションを数回繰り返すころがある．TCP コネクションの試行回数に応じて遅延を増加させる方法だと，正規の利用者にも大きな遅延をかける可能性がある．このため，しきい値を導入する．TCP コネクション試行回数のしきい値を定め，それを超えない回数までは遅延の増加量を小さなものとする．TCP コネクション試行回数がしきい値を越えたならば，scan 攻撃だと判断し，遅延の増加量を大きくしていく．一定時間，TCP コネクションが志向されなかったものについては、その試行回数を 0 にリセットする．

3.2.2 多くの TCP コネクションの試行を必要とするプロトコル

プロトコルの中には，多くの TCP コネクションを利用するものもある．HTTP(80 番)などが該当する．これらのプロトコルに関しては，white list を使用し，指定されたプロトコルに対して遅延を行わないようにすることで解決を図る．white list には throttling の対象としないあて先ポート番号を記録する．TCP

コネクションのあて先ポート番号が white list に該当する場合には throttling を行わない.

3.3 遅延アルゴリズム

以上により, 遅延アルゴリズムは以下のようになる.

(1) パケットを 1 つ受け取る.

(2) パケットの送信元アドレス, 宛先アドレス, 宛先ポートのいずれかが white list にある, あるいは, プロトコルフィールドが TCP 以外のときは, 遅延なしキューに送り, (1)へ戻る.

(3) パケットが TCP コネクション開始パケット (SYN=1, ACK=0)のとき, 送信元アドレス, 宛先アドレスそれぞれの試行回数, 遅延時間を更新する.

(4) 送信元アドレスに対する遅延時間, 宛先アドレスに対する遅延時間を検索し, その和をこのパケットの遅延時間とする.

(5) 送出時刻＝現在時刻＋遅延時間を計算し, パケット自身とともに遅延キューへ送り, (1)へ戻る.

4. 実装

4.1 全体構成

システム全体の流れを図 2 に示す. システムは一方のネットワークに流れるパケットを全て受信し, もう一方のネットワークに送信する. ただし, 送信の前に, 受信したパケットの解析を行い, throttling を行うかどうかを判断する. throttling を行う場合は, パケットに遅延をかけた後に送信を行う.

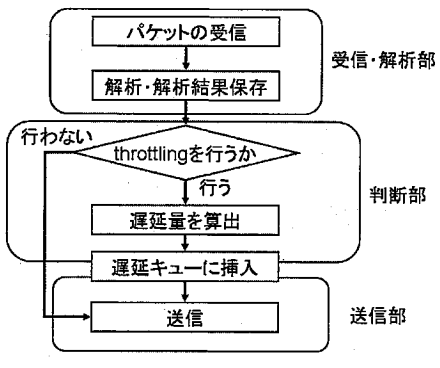

図 2　システムの流れ

4.2 受信・解析部

ここでは受信したパケットに throttling を行うかを判断するために, パケットを解析し, 解析結果を保存しておく.

まず、パケットを受信し, TCP ヘッダ, IP ヘッダの解析を行う. 本システムで使用するものは, 送信元 IP アドレス, 宛先 IP アドレス, 宛先ポート番号, TCP コントロールフラグビット, 到着時刻の情報である. 解析結果の保存では送信元 IP アドレス毎および宛先 IP アドレス毎に到着時刻と TCP コネクション試行回数を保存していく(図 3). 到着時刻は最終アクセス時刻として使用する. TCP コントロールフラグビットの SYN フラグが 1, ACK フラグが 0 である場合に, TCP コネクションが試行されたものとしてカウントを行う. 既に IP アドレスの情報が保存されている場合には, 回数, アクセス時刻の更新を行う.

解析結果をいつまでも保持していると, 正規のユーザの TCP コネクションの試行回数がしきい値を超えるという問題が発生する. そのため, 定期的に保存している情報を整理する. 送信元 IP アドレスにおいては, 一定周期で, 保存されている各 IP アドレスの最終アクセス時間と現在時間を比較し, 一定時間更新が無いアドレスの情報を破棄する. 宛先 IP アドレスにおいては一定周期で全ての情報を破棄する.

送信元IPアドレス別の情報

IPアドレス	TCPコネクション試行回数	最終アクセス時間(到着時間)
133.37.xxx.aaa	16	20061010 03:00:56
133.37.yyy.bbb	8	20061010 03:48:33
133.37.zzz.ccc	5	20061010 03:50:12

⋮

宛先IPアドレス別の情報

IPアドレス	TCPコネクション試行回数	最終アクセス時間(到着時間)
133.37.aaa.xxx	10	20061010 03:48:20
133.37.aaa.yyy	7	20061010 03:50:33

図 3　解析結果の保存

4.3 判断部

判断部では, 受信・解析部で得られた情報から throttling を行うかを判断する. throttling を行わないのは, 次の3つの場合である.

＜throttlingを行わない場合＞
(1)パケットがTCPを使用していない
(2)ポートあるいは送信元，あて先 IP アドレスが，white listに該当する
(3)宛先IPアドレスのTCPコネクション試行回数と送信元 IP アドレスの TCP コネクション試行回数がともに0回である

throttlingを行わない場合は、パケットを送信部が持つ遅延なしキューに挿入する。

throttlingを行う場合は，まず遅延時間の計算を行う．遅延時間は，宛先IPアドレスのTCPコネクション試行回数と送信元 IP アドレスの TCP コネクション試行回数からそれぞれ計算された値の合計となる．それぞれの計算時間は次のとおりである．

遅延時間＝試行回数×α
(試行回数が閾値未満のとき)

遅延時間＝(試行回数－閾値)×β＋閾値×α
(試行回数が閾値以上のとき)

閾値，α，βの現在の値を表 1 に示す．

次に，得られた遅延時間を現在時刻に加算して送信時刻を求める．最後に，パケットに送信時刻を設定し，送信部が持つ遅延キューにパケットを挿入する．

表 1 各パラメータ値

	閾値	α	β
送信元アドレス	16	1.0	0.1
宛先アドレス	256	0.1	0.0

4.4 送信部

送信部はパケットの送信を行う．送信するパケットはthrottlingを行うパケットと，行わないパケットの2つに分けられる．それぞれのパケットに対する送信部の働きについて述べる．

4.4.1 throttlingを行わない場合

遅延なしキューからパケットを取り出し、すぐに送信する．

4.4.2 throttlingを行う場合

パケットに遅延をかける必要があるため，システム内でパケットを保持する機能が必要となる．このため，パケットを保持する遅延キューを持つ．遅延キューには，送信時刻が設定されたパケットが挿入される．この挿入は判断部が行う．各パケットは送信時刻が早い順に並ぶように挿入される．送信部は遅延キューの先頭の送信時刻と現在の時刻を比較し，送信時刻をすぎているパケットを遅延キューから取り出し，送信を行う(図 4)．

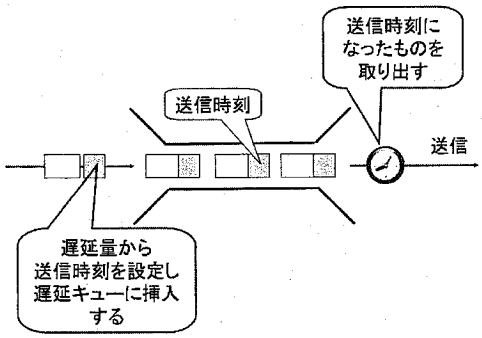

図 4 遅延キュー

5. 性能評価実験

性能評価実験として，実験環境を作成し，scan攻撃にどの程度の効果があるかを確認した．

5.1 実験方法

攻撃用コンピュータを一方のネットワークに配置し，もう一方のネットワークに攻撃対象とするコンピュータ設置し，その間に本抑制システムを配置する．そして，ポートスキャンツールである nmap[3]を用いてscan 攻撃を行う．次に，本システムを外し，再度scan攻撃を行う．実験環境を図6に示す．

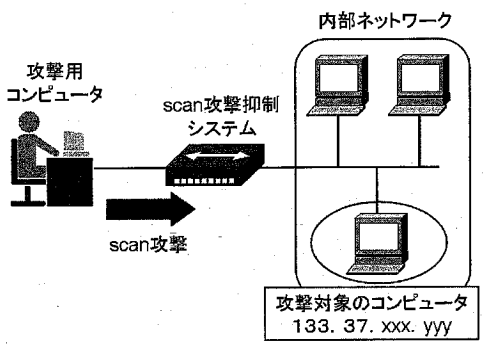

図 5 実験用ネットワーク

5.2 実験結果

実験結果のログを以下に抜粋する．ログの分量が多いため，必要のない部分は省略してある．

```
Starting Nmap 4.11
Initiating SYN Stealth Scan against
example.csis.oita-u.ac.jp (133.37.xxx.yyy)
[1680 ports] at 02:07
Discovered open port 80/tcp on 133.37.xxx.yyy
Discovered open port 22/tcp on 133.37.xxx.yyy
Discovered open port 21/tcp on 133.37.xxx.yyy
SYN Stealth Scan Timing: About 18.07% done; ETC:
02:10 (0:02:30 remaining)
Discovered open port 2601/tcp on 133.37.xxx.yyy
SYN Stealth Scan Timing: About 44.61% done; ETC:
02:14 (0:03:52 remaining)
SYN Stealth Scan Timing: About 45.74% done; ETC:
02:22 (0:08:23 remaining)
SYN Stealth Scan Timing: About 59.54% done; ETC:
02:54 (0:19:11 remaining)
Warning: Giving up on port early because
retransmission cap hit.
SYN Stealth Scan Timing: About 76.66% done; ETC:
03:09 (0:14:26 remaining)
Discovered open port 111/tcp on 133.37.xxx.yyy
SYN Stealth Scan Timing: About 98.70% done; ETC:
03:25 (0:01:00 remaining)
The SYN Stealth Scan took 4700.06s to scan 1680
total ports.
Host example.csis.oita-u.ac.jp
(133.37.xxx.yyy) appears to be up ... good.
Nmap finished: 1 IP address (1 host up) scanned
in 4700.315 seconds
Raw packets sent: 2426 (106.742KB) | Rcvd: 2432
(111.982KB)
```

図 6. システム使用時の結果

```
Starting  Nmap 4.11
Initiating SYN Stealth Scan against
example.csis.oita-u.ac.jp (133.37.xxx.yyy)
[1680 ports] at 04:30
Discovered open port 22/tcp on 133.37.xxx.yyy
Discovered open port 80/tcp on 133.37.xxx.yyy
Discovered open port 21/tcp on 133.37.xxx.yyy
Discovered open port 2601/tcp on 133.37.xxx.yyy
Discovered open port 111/tcp on 133.37.xxx.yyy
The SYN Stealth Scan took 1.58s to scan 1680
total ports.
Host example.csis.oita-u.ac.jp
(133.37.xxx.yyy) appears to be up ... good.
Nmap finished: 1 IP address (1 host up) scanned
in 1.820 seconds
Raw packets sent: 1683 (74.050KB) | Rcvd: 1679
(77.230KB)
```

図 7. システム非使用時の結果

システム非使用時は 1.820 秒で scan 攻撃が終了しているのに対し，システム使用時は，攻撃終了まで 4700.315 秒の時間が経過している．終了までに 2500 倍以上の時間がかかっている．攻撃者が scan の終了を待たずに攻撃を断念する可能性は十分考えられる．またシステムが無い場合に比べて，不正アクセスの前に対策を行える可能性は高いと考えられる．

6. まとめ

本論文では，throttling を利用した scan 攻撃抑制システムについて述べた．scan 攻撃を抑止することで，scan 攻撃につづく不正アクセスを予防できると考えられる．本システムでは，TCP コネクションの開始を検知して throttling を行うため，FIN パケットや PUSH パケットを送りつける XmasTree scan 攻撃などに対応していない．これらの攻撃に対しても throttling を行えるよう判断部の改良が必要である．今後は，本システムを大学 LAN 内で運用させ，有効性の検証を行なうとともに，保存している試行回数等の情報をリセットする頻度，遅延時間計算のためのパラメータの調整等を行っていきたい．

参考文献

[1] 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
[2]IPA コンピュータ不正アクセス被害防止対策集, http://www.ipa.go.jp/security/ciadr/cm01.html#DoS
[3]nmap http://insecure.org/nmap/