2009-IOT-4 (35)
2009 / 3 / 5

TCP コネクション確立の偽装と その計数による scan 攻撃検知について

大塚賢治[†] 兒玉清幸[†] 衣笠雄気[††] 吉田和幸[†††]

[†]大分大学大学院工学研究科[††]大分大学工学部[†††]大分大学学術情報拠点情報基盤センター

サーバの使用状況や動作しているサービスの調査を行う scan 攻撃が後を絶たない。scan 攻撃の場合，宛先のアドレスをランダムに設定しコネクション要求を送るため、応答がないことが多い。このため、存在しない IP アドレスに対してコネクション要求を行なう回数を数えることで scan 攻撃を検知することができる. しかしながら, 検知した IP アドレスを単純にファイアウォールなどで止めた場合, TCP half open 攻撃のように送信元の IP アドレスを偽装する可能性が高い攻撃に対して，問題が起こる可能性がある．そこで，TCP コネクション要求に対して送信元アドレスが偽装されていないか確認するとともに、コネクションが確立したか否かで scan 攻撃を検知するシステムを試作した．本稿では、攻撃検知手法と送信元の確認の効果について述べる

キーワード IDS，ネットワークセキュリティ

Scan Attack Detection using the Counting of TCP Connection Request with Pseudo SYN-ACK Reply.

Kenji OTSUKA[†] Kiyoyuki KODAMA[†] Yuuki KINUGASA[†] Kazuyuki YOSHIDA[††]

[†]Department of Computer Science and Intelligent Systems, Oita University

[††]Center for Academic Information and Library Services, Oita University

There are a lot of scan attacks which look for state of the server or check on service. Scan attacker send TCP connection request to random destination address, so there are seldom answer for them. For this reason, we can detect scan attack by count the number of failed connection request. However if we refuse detected IP address with firewall etc, a problem may occur for attacks like TCP half open attack with fake source IP address. We implement the system which detected scan attacks that we confirm source IP address is not camouflaged for TCP connection demand, and connection successfully or not establishes. In this paper, we describe this attack detection technique and its effect.

Keyword IDS, network security

1. はじめに

セキュリティホールの残っているホストや，特定のサービスを行っているホストを探す 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．scan 攻撃によるホストの発見後，ホストへの攻撃が行われる可能性がある．この攻撃により，サービスを行えない，不正アクセスが行われるなどの問題が発生することが考えられる．不正アクセスによる侵入を許した場合，他のホストへの攻撃の踏み台，spam メールの中継，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ことが考えられる．

この結果を受け，単位時間当たりのコネクション要求回数から攻撃者を判断するシステムを作成し，運用を行った[5]．しかし，コネクション要求回数から攻撃者を判断する場合，攻撃者を決定する基準は，HTTP などのコネクション確立と切断を繰り返す動作をするプロトコルでは大きめに設定するなど，プロトコルごとに設定する必要がある．また，大量にコネクション要求を送ってくる攻撃の一種である half open 攻撃では送信元の IP アドレスを偽装している可能性もあり，検知した IP アドレスをファイアウォールでフィルタする等，対処した際に問題が出る可能性がある．

scan 攻撃では非常に多くの宛先に向かってコネクション要求のパケットを送信してくるため TCP コネクションが確立できないものが多くなる．この特徴を利用し，TCP コネクションが確立するか否かにより scan 攻撃を判断するシステムを試作した．また，送信元 IP アドレスが偽装されている場合を考え，コネクション要求に対する内部からの応答がない場合，システムが TCP 接続確立の手順通り応答を偽装しそれに対する攻撃者の反応により存在を確認する機能を実装した．

2. 関連研究等

フリーソフトの IDS(Intrusion Detection System)である snort[2]では、プリプロセッサにより scan 攻撃の検知を行っている．ステルススキャンと呼ばれる，ホストのログに残らない scan 攻撃の検知が可能である．しかし，scan 攻撃の判断基準が「単位時間当たりのパケットの送信回数のみ」であるため，Web クローラやプロキシを使用している場合，アクセス回数が閾値を超えてしまい誤検知が増えてしまうことや，攻撃者がパケットの送信時間間隔を大きくすることで容易に回避を行えるという短所がある．

また，Bro IDS[3]は，コネクションの状態を監視し，送信したパケットに対する応答がないものまたは拒否を行なったものを計数する．その計数値が閾値を超えた場合に scan 攻撃だと判断する．このため snort に比べて誤検知は少ない．しかし TCP half open 攻撃，smurf 攻撃等の送信元 IP アドレスを偽装する攻撃に対して偽装された送信元を攻撃者と誤検知する可能性がある．Bro IDS は IDP(Intrusion Detection and Prevention system)の機能も持っているおり，設定によりアクセスを禁止することもできる．

3. 攻撃者検知システム

3.1. システムの概要

本システムは外部ネットワークから scan 攻撃を行なうホストの検知とそのホストの存在の確認を行なうものである．大分大学が送受信するパケットすべてをミラーリングし，本システムの入力としている．対象とする scan 攻撃は TCP を用いたものを対象とする[4]．

一般ユーザがインターネットへアクセスする場合，外部へ公開されている HTTP サーバやメールサーバなどへのアクセスを行なうため，コネクション確立が失敗することや，短期間に大量のコネクション要求を送信してくることは少ない．

しかし，scan 攻撃を行なってくる攻撃者は内部ネットワークのホストの使用状況やサービスの稼動状態などの情報を収集するため，パケットを非常に多くの宛先に向かって送信し，それに対する応答により内部ネットワークの情報を収集しようとする．このため，コネクション要求を用いた scan 攻撃であれば，パケットの送信数に対して，コネクション確立ができる数は少なくなり，FINScan などコネクション確立を行なわず調査する scan 攻撃であれば，未解決のコネクション数が増加する．また，短期間で情報を得ようとするため短期間に大量のパケットを送信してくることもある．

これらの特徴より，コネクション確立が成功したか否かあるいは，短時間に大量のコネクション要求パケットを送信してくるかで scan 攻撃を検知する．

3.2. 送信元ホストの確認

本システムは，ホストが存在しない，または，使用していないポートへ TCP フラグオプションの SYN フラグを 1 としたパケットを送信してきたホストに対し，SYN/ACK フラグを 1 としたパケットを送信する機能

を持つ．代理応答により，送信元ホストから応答となる ACK フラグが 1 となっているパケットが返信されれば，送信元が存在していると確認を行なうことができる．|また，返信がなければ送信元は IP アドレスが偽装されていると判断できる[6].

3.3. scan 攻撃の検出

3.3.1設定情報

本システムでは，設定と情報を保持するためにリストを使用する．外部から来たパケット送信元ホストの IP アドレスなどの情報を保持する送信者リストと，scan 攻撃を行なっていると判断された攻撃者の情報を保持する攻撃者リスト，特定のポート番号を除外する除外リストである．

送信者リスト，攻撃者リストはシステムの起動後，動的に作成され，除外リストはシステム起動前に静的に作成される．

以下簡単に説明を行う．

- **送信者リスト**

送信元 IP アドレスごとに 1 秒間のコネクション要求回数，未解決コネクション数を保持する．また，送信元 IP アドレスのコネクションごとに送信元ポート番号，宛先 IP アドレス・ポート番号，シーケンス番号，コネクションの状態, パケット受信時刻, を保持する．

コネクション確立後，FIN フラグまたは RST フラグを使用してコネクションを切断した場合，リストから削除される．しかし，コネクション確立なしで送信してきたパケット数と代理応答を行なった回数の合計，または単位時間当たりのコネクション要求回数が閾値を越えた場合，攻撃者リストへと登録され，送信者リストから削除される．

- **攻撃者リスト**

攻撃者と判断されたホストを登録する．登録する情報は送信元 IP アドレスのみである．

- **除外リスト**

外部からのコネクションをファイアウォールで止めているなど検査を行なわないでよいポート番号を登録する．

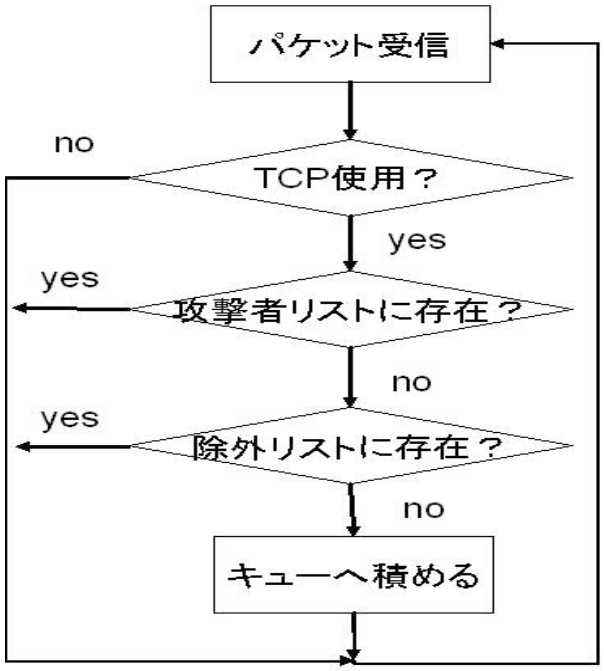

図 1　受信時のパケット解析

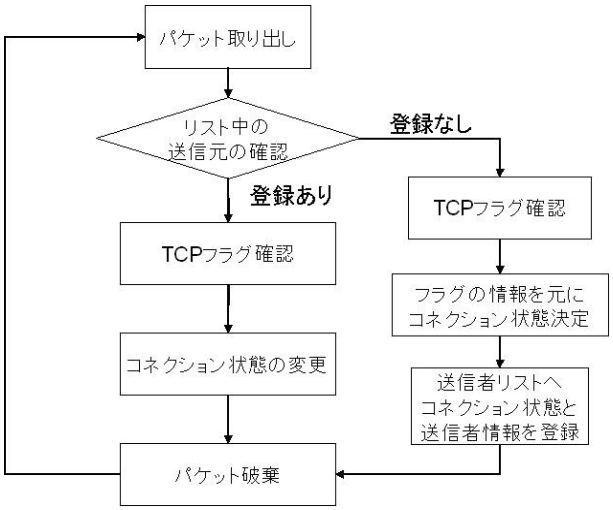

図 2　キューからのパケット取り出し

3.3.2検出方法

外部ネットワークから受信したパケットについて，(1)ＴＣＰを使用しているか, (2)攻撃者リストに存在せず，除外リストにも存在しないものを抽出し，キューへと積める．それ以外はパケットを破棄する(図 1).

次に，キューからパケットをひとつ取り出し，そのパケットの解析を行う(図 2).

送信者リストに登録のない状態で SYN フラグが 1 のパケットを送信された場合，送信者リストに登録される．それ以外であれば未解決コネクション数を増加した後，登録する．

表 1　パケットによる状態変化

TCPフラグオプション	コネクション状態				
	登録なし	接続確認	接続確立	終了	未解決
SYN	接続確認				
FIN	未解決	未解決	終了		
RST	未解決	未解決	終了		
ACK	未解決	接続確立			

＊斜線部は状態が変化しないことを示す

送信者リストに登録されていれば，送信者リストに保持しているコネクション状態とパケットの TCP フラグオプションを元に，コネクション状態を変更する(表 1)．表 1 では上の行に存在する TCP フラグオプションの優先度が高く設定されている．そのため，SYN/ACK パケットのように複数のフラグオプションが 1 となり送信されてきた場合，優先度の高い方のフラグオプションと同じ扱いを受ける．

コネクション要求を送信してきたが内部ネットワークにホストが存在しないなど応答がない場合，システムが代理応答を行う．その代理応答に対して確認応答がある場合，攻撃者が存在するものとして未解決コネクション数を加算する．応答がない場合，攻撃者の IP アドレスが偽装されているとして未解決コネクション数を変化させない．

最後にシステムは一定時間ごとに送信者リストに保持されている情報の走査を行う．この際に，未解決コネクション数または 1 秒間のコネクション要求回数が閾値を超えているかの確認を行なう．閾値を超えていなければリストに保持してある送信元ＩＰアドレスの情報の表示を行なう．どちらかが超えている場合，攻撃者リストに情報を登録し，送信者リストから情報を削除する(図 3).

3.5 運用環境

システムの使用している PC の OS とハードウェア性能および LAN スイッチは以下のとおりである．

OS　：　Red Hat Linux 2.4.20-8

CPU　：　Intel(R) Xeon(TM) CPU 3.06GHz

メモリ　：　2Gbyte

スイッチ　：　DELL PowerConnect 3324

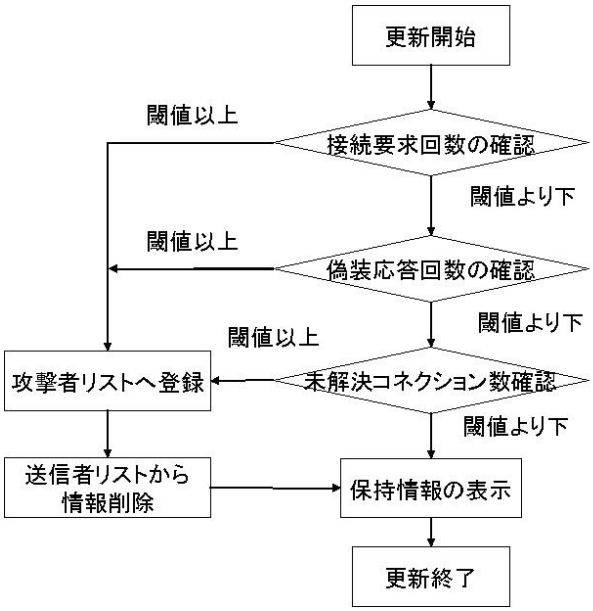

図 3　更新処理

また，除外リストとして以下のポート番号を除外している．

80,135～139,445,443

これらのポートは Netbios など，LAN 内部などで使用するものなので，LAN 外部からのすべてのアクセスをファイアウォールなどで除外することができる．また，HTTP サーバ以外への 80・443 ポートへのアクセスを禁止することで容易に禁止できるため除外した．

4.　運用結果

4.1.　ログの収集

本システムが動作している PC で同時にパケットキャプチャツールである tcpdump を用いて TCP が使用されているパケットのみを収集した．

収集を行なった期間は 2009 年 2 月 7 日 23 時 20 分から 2009 年 2 月 8 日 9 時 20 分の 10 時間である．

4.2.　攻撃者の決定について

今回は試験的にシステムを作成したため，未解決コネクション数が 5 回を超えたもの，または 1 秒間のコネクション要求回数が 30 回を超えたものを攻撃者として判断する．

表 2　攻撃者検知数

検知数	569件
送信元の存在確認	408件
送信元の存在未確認	161件

4.3.　データの検証

システムを運用し, 検知した攻撃者数が表 2 となる．まず，表 2 中の攻撃者のうち，代理応答に対し，応答を返したため送信元の存在が確認できたものが「送信元の存在確認」となっている．また，コネクションを張らずにパケットを送信してきた，または，代理応答に対して RST パケットを送信してきた送信元を「送信元の存在未確認」となっている．

まず，この結果について見ていく．システムのログを図 4 に示す．state：EXIST となっているものが攻撃者から代理応答に対する応答があったものであり，state:ABSENT となっているものは応答がなかったものとなる．図 5 から，本来 5 回で検知できる攻撃者が state:ABSENT と判断されることで判断に回数がかかるようになってしまっていることがわかる．これは，接続要求に対するホストからの SYN/ACK パケットによる応答があるか判断を行なう時間の間に，攻撃者側のポートが閉じているためと考えられる．

次に,「存在未確認」となっている攻撃者の検証を行なう．これに該当する送信元には，コネクションを確立せずにパケットを送信してきた，または，代理応答に対して RST パケットを送信してきた, のどちらかであった．検証を行なった結果，71 件がコネクションを確立せずにパケットを送信している送信元であり，残りの 90 件が代理応答に対して RST パケットを送信している送信元であった．

コネクションを確立せずにパケットを送る送信元に対して代理応答を行なうにしても, 本来が RST パケットを送信するか，何も送信しないため，送信元の確認がうまく行なえない．また，代理応答に対して RST パケットを送信してくる送信元では送信元 IP アドレスが偽装されていて, 本来送信していない IP アドレスに題して代理応答をしている可能性がある．

RST パケットを送信する送信元は主に DNS へのアクセス，メールサーバへのアクセス，または，短期間で集中的にパケットを送信してくるといったものがあった．特に，DSN へのアクセスは SYN/ACK パケットの後に RST パケットを送信する送信元だけであった(図 5).

最後に，代理応答を行なったアドレスに対し，同一ネットワークではない複数のアドレスから複数回のアクセスがされていることがわかった(図 6). 図 6 に示したものは一部を抜粋したものであるが，実際にはひとつの送信元から 10 数回のアクセスを受けている. これは，ボットネットなど，反応のあったホストに対し，感染範囲を広げる，または攻撃を行なうために接続を行うとしているのではと考えられる．そのため，代理応答をうまく利用することで，scan 攻撃と判断するまでのアクセス回数を減らすことができるであろう．

5.　まとめと今後の課題

コネクション確立の有無により scan 攻撃を判断するシステムを試作し，コネクション要求を未使用のアドレスまたはポートに対し送信してきた場合，代理応答を行なう機能を実装した．

実験の結果，scan 攻撃の検知は行なえていることがわかった．また，代理応答を利用することで，検知までのアクセス回数を減らすことができるのではないかと考えられる．

今後の課題として，代理応答をより確実に行い攻撃者の判断を正確にすることで送信元 IP アドレスを攻撃者と判断できるまでの数を減らす．コネクションの確立を行なわないでパケットを送信してくる scan 攻撃に対する対処や，代理応答に対する RST フラグが 1 のパケットを送信された際の対処についてといったことが挙げられる．現状，コネクションを確立せずにパケットを送信してくる送信元の確認は行なえていない. また，RST パケットを送信してくる送信元は RSTScan を行なっているとして判断している．しかし，IP アド

```
attack_sip:58.63.148.144
        sp:3932->dip:133.37.  .30 dp:54060 state:EXIST Sun Feb  8 06:29:56 2009
        sp:3713->dip:133.37.  .30 dp:54060 state:EXIST Sun Feb  8 06:27:11 2009
        sp:3532->dip:133.37.  .30 dp:54060 state:EXIST Sun Feb  8 06:24:48 2009
        sp:3397->dip:133.37.  .30 dp:54060 state:ABSENT Sun Feb  8 06:23:06 2009
        sp:4349->dip:133.37.  .30 dp:54060 state:EXIST Sun Feb  8 05:40:44 2009
        sp:4196->dip:133.37.  .30 dp:54060 state:EXIST Sun Feb  8 05:38:26 2009
        sp:4013->dip:133.37.  .30 dp:54060 state:EXIST Sun Feb  8 05:36:15 2009
        sp:3874->dip:133.37.  .30 dp:54060 state:ABSENT Sun Feb  8 05:34:34 2009
        sp:3699->dip:133.37.  .30 dp:54060 state:ABSENT Sun Feb  8 05:31:38 2009
```

図 4　システムログ

```
attack_sip:66.238.93.161
        sp:2287->dip:133.37.  .133 dp:53 state:RSTSCAN Sun Feb  8 00:33:31 2009
        sp:2282->dip:133.37.  .133 dp:53 state:RSTSCAN Sun Feb  8 00:33:31 2009
        sp:2272->dip:133.37.  .133 dp:53 state:RSTSCAN Sun Feb  8 00:33:31 2009
        sp:2544->dip:133.37.  .133 dp:53 state:RSTSCAN Sat Feb  7 23:55:10 2009
        sp:2537->dip:133.37.  .133 dp:53 state:RSTSCAN Sat Feb  7 23:55:10 2009
        sp:2525->dip:133.37.  .133 dp:53 state:RSTSCAN Sat Feb  7 23:55:10 2009
```

図 5　DNS へのアクセス

```
attack_sip:77.202.182.253
        sp:3632->dip:133.37.  .83 dp:15729 state:EXIST Sun Feb  8 09:10:28 2009
        sp:2307->dip:133.37.  .83 dp:15729 state:ABSENT Sun Feb  8 08:29:25 2009
attack_sip:80.34.139.78
        sp:17158->dip:133.37.  .83 dp:15729 state:EXIST Sun Feb  8 08:43:14 2009
        sp:16276->dip:133.37.  .83 dp:15729 state:ABSENT Sun Feb  8 08:19:30 2009
attack_sip:81.9.148.21
        sp:1504->dip:133.37.  .83 dp:15729 state:EXIST Sun Feb  8 04:16:59 2009
        sp:4393->dip:133.37.  .83 dp:15729 state:ABSENT Sun Feb  8 04:00:42 2009
attack_sip:81.32.109.85
        sp:4693->dip:133.37.  .83 dp:15729 state:EXIST Sun Feb  8 08:29:01 2009
        sp:3824->dip:133.37.  .83 dp:15729 state:EXIST Sun Feb  8 08:08:23 2009
attack_sip:81.43.98.228
        sp:18814->dip:133.37.  .83 dp:15729 state:EXIST Sun Feb  8 04:22:00 2009
        sp:15099->dip:133.37.  .83 dp:15729 state:EXIST Sun Feb  8 03:50:30 2009
attack_sip:81.92.178.47
        sp:46715->dip:133.37.  .83 dp:15729 state:EXIST Sun Feb  8 08:22:14 2009
        sp:40874->dip:133.37.  .83 dp:15729 state:ABSENT Sun Feb  8 07:13:10 2009
attack_sip:83.40.73.195
        sp:1406->dip:133.37.  .83 dp:15729 state:EXIST Sun Feb  8 03:16:33 2009
        sp:3736->dip:133.37.  .83 dp:15729 state:EXIST Sun Feb  8 02:31:40 2009
```

図 6　代理応答したアドレスへのアクセス

レスが偽装されていて，RST パケットを送信している可能性もあるため, scan 攻撃か正規の RST パケットなのか判断する必要がある．

最後に，代理応答を行なうことでボットネットなどからの攻撃を集めている可能性がある．そのため，代理応答を利用することで，さらに少ない回数での攻撃者の判断ができるのではと考えられる．

参考文献

[1]　大塚賢治, 兒玉清幸, 吉田和幸, “偽装応答による scan 攻撃抑制システムについて”, マルチメディア，分散，協調とモバイル(DICOMO2008)シンポジウム pp.182-118, 2008.7

[2]　Snort：http://www.snort.org

[3]　Bro：http://www.bro-ids.org

[4]　J.Postel, “Transmission Control Protocol”, RFC 793, Sep 1981

[5]　衣笠雄気, 大塚賢治, 兒玉清幸, 吉田和幸, “アクセス制御を用いた scan 攻撃抑制システムについて”, 電気関係学会九州支部連合大会(第 61 回連合大会)講演論文集 11-2P-13,2008/9/24

[6]　兒玉清幸, 大塚賢治, 南浩一, 吉田和幸, “偽装応答を用いた scan 攻撃抑制システムの提案”, FIT2007(第６回情報科学技術フォーラム)講演論文集　pp.71-73, 2007