TCPコネクション要求回数の計数による攻撃者の検知

衣笠雄気† 大塚賢治†† 兒玉清幸†† 吉田和幸†††
†大分大学工学部 ††大分大学大学院工学研究科 †††大分大学学術情報拠点情報基盤センター

SSHサーバ等を探し，そのサーバを攻撃するため，大量のTCPコネクション要求が送られてくる．SSHサーバなどが発見されてしまうと攻撃者は不正侵入を試み，不正侵入を許してしまうと他のホストへの攻撃の踏み台になるなどの問題がある．我々はTCPコネクション要求に対して遅延・偽装応答を行うscan攻撃抑制システムを提案し，運用してきた．運用の結果，攻撃者は単位時間当たり多数の接続要求を送信し，短期間でIPネットワークアドレス中のサーバの探索を行っていることがわかった．そこで，単位時間当たりのTCP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提案する．本論文では，単位時間当たりのTCPコネクション要求回数から攻撃者を判断するシステムの構築とその運用結果について述べる．

Detecting Attacks by Counting TCP Connection Requests

YUUKI KINUGASA† KENJI OTSUKA† KIYOYUKI KODAMA† 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 packets to search and to attack servers. Once, the servers are discovered, attackers attempt to make intrude into the servers. If attacker intrudes into the servers, there is a problem such as becoming the step ladder of the attack to other hosts. To prevent scanning attacks, we have suggested and implemented a system which delays attacking packets or makes pseudo-responses against attacking TCP connection requests. As a result of the operation of that system, we found that a lot of TCP connection requests per unit time to search servers in the IP network address in a short time. So, we suggest a system which identifies attackers by the number of their TCP connection request per unit time. In this paper, we describe the system and results from its operation.

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．

そのため，ネットワーク管理者はパケットのフィルタリングなど，細かい設定を行うことが必要になる．しかし，大学などでは，ホストやユーザの数が多く，外部からPCが持ち込まれることが考えられる．厳しい設定をネットワークの境界に行った場合，内部ネットワークのPCの使用に問題が出ることが考えられ，対策は難しい．

そこで，ネットワーク単位で攻撃を抑制し，TCPコネクション接続要求に対し，偽装応答や遅延をかけるthrottlingを用いたscan攻撃抑制システムを開発・運用を行った[1]．運用の結果，攻撃者は単位時間当たり多数の接続要求を送信し，短時間でネットワーク内のIPアドレス全体の状態の探索を行っていることがわかった(図1)[1]．この結果を受け，単位

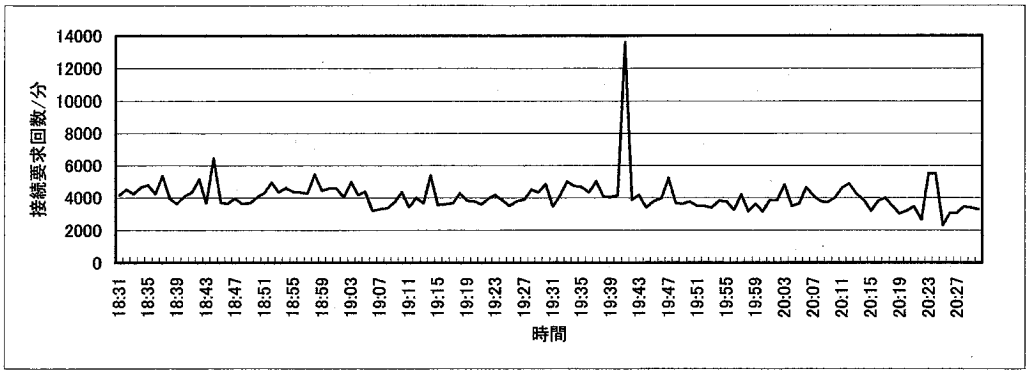

図1 接続要求回数

時間当たりの接続要求回数から攻撃者を判断するシステムを作成し，試験的に運用を行った．

2. 関連研究等

インターネットとLANとの間に設置し，scan攻撃等を検出，抑制するシステムとしてフレッツ・セーフティ[2]やpsad(port scan attack detector)[3]等がある．フレッツ・セーフティは対応機器をルータとPCとの間に設置することで，対応機器と直接繋がっているPCがscan攻撃やDoS攻撃を受けたとき攻撃者を検知する．フレッツ・セーフティは小規模ネットワークを保護対象としており，ポートスキャンばかりでなく，ウィルス検知等も行う．なお，フレッツ・セーフティは平成20年3月31日にサービスの提供を終了している．psadはポートスキャンの範囲や開始時刻，終了時刻を表示するといった有用性の高い機能がある．psadは保護ネットワーク内のそれぞれのホストに対するアクセス回数がしきい値を超えた場合に攻撃者を検知する．これらのシステムではネットワーク全体から攻撃対象を探すネットワークスキャンに対して，その検知が比較的弱い．

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接続要求を行ってきたIPアドレスが攻撃者であるかどうか判断するものである．対象とするscan攻撃はTCP[4]によるコネクション接続要求のみである．

攻撃者は短期間で大量の接続要求を行い内部ネットワークのIPアドレスの使用状況を調査する．その結果を受けて，単位時間当たりのコネクション接続要求回数から攻撃者を判断する．

一般のユーザは短時間に大量のTCPコネクション接続要求を送信してこない．そのことから外部ネットワークの同一IPアドレスから行われるTCPの接続要求の回数が単位時間中にしきい値を超えた場合，攻撃者であると判断する．

3.2. 設定情報と検出結果

本システムでは，外部ネットワークからのアクセス情報を保持するためリストを使用する．接続要求を行ってきたIPアドレスを保持する送信元リスト，外部ネットワークからの接続要求であっても安全であると保証されているネットワークアドレスを保持するホワイトリストがある．

送信元リストはシステムの起動後，動的に作成される．一方，ホワイトリストはシステムの起動前，静的に作成される，

以下簡単に説明を行う．

- **送信者リスト**

内部ネットワークに対して接続要求を行ってきたパケットの送信元IPアドレスと，そのIPアドレスからの

接続要求回数が登録され，しきい値を超えたものが出力される．接続要求回数は一定時間ごとに初期化される．

- ホワイトリスト

外部のネットワークであっても安全であると保障されているネットワークアドレスをあらかじめ設定ファイルにより登録する．ホワイトリストを用いる理由は，攻撃者ではない IP アドレスからの接続要求であっても攻撃者であると判断してしまうことを防ぐためである．

3.3. scan 攻撃の検出

システムは外部ネットワークから来たパケットを受信し，受信したパケットが TCP コネクション接続要求である場合，接続要求を行ってきた IP アドレスがホワイトリストに存在するネットワークアドレスからのものかどうかを調べ，存在する場合には正常な通信として処理を終了し，次のパケットの解析に移る．そうでないとき，接続要求を行ってきた IP アドレスが送信者リストに存在するかどうかを調べ，送信者リストに存在する場合には接続要求回数に1を加算し，存在しない場合には送信者リストに IP アドレスを登録し接続要求回数の値を1にする(図 2)．さらにシステムは設定された一定時間ごとに送信者リストの内容を走査し，ある IP アドレスからの接続要求の回数が設定したしきい値を超えていた場合，その IP アドレスは攻撃者であると判断し IP アドレスと単位時間当たりの接続要求回数を出力する．出力した後，接続要求回数を0に初期化し次の期間中の接続要求回数を計数し始める．1回でもしきい値を超えて攻撃者と判断された IP アドレスはその後，しきい値を越えていなくても IP アドレスと各期間の接続要求回数を出力する(図 3)．今日，運用環境で用いているスイッチには IP アドレスによるフィルタのみサポートされているためパケットを収集後，接続要求のみ抽出している．スイッチのアクセス制御リストが TCP flag の値によりフィルタする機能があれば，それを使用することで効率よくパケットを収集できるであろう．

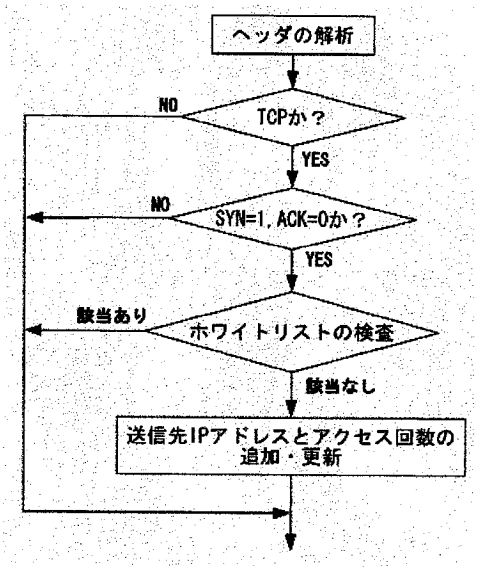

図 2 パケットの解析

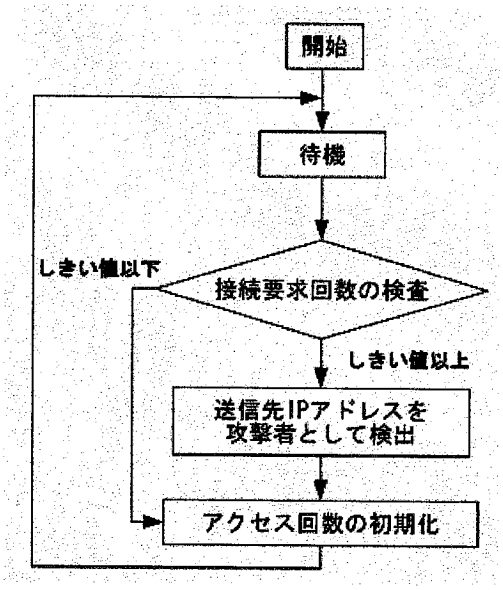

図 3 攻撃者の表示

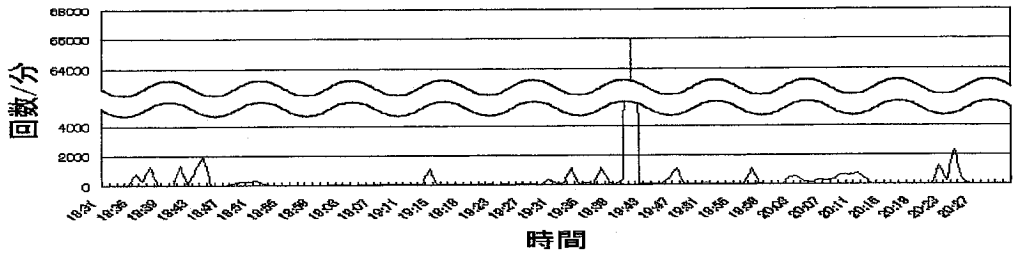

図 4 システムの接続要求回数

3.4. 運用環境

大分大学に入ってくるパケットすべてをミラーリングして，本システムの入力としている．

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になっている．

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

4. 運用結果

4.1. ログの収集

本システムが動いている 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 フラグの SYN のみが1となっているパケットのみを収集した．

収集を行った期間は 2008 年 9 月 9 日の午後 6 時 31 分から午後 8 時 31 分までの 2 時間と，2008 年 10 月 22 日午後 1 時 42 分から 2008 年 10 月 24 日午後 1 時 42 分までの 48 時間である．

4.2. 攻撃者の決定について

2008 年 9 月 9 日の午後6時31分からの2時間はデータをより多く採集するため収集間隔を1分間，しきい値を10回とし，2008年10月22日午後1時42分からの48時間は収集間隔 1 分しきい値100回とした．

4.3. データの検証

2008 年 9 月 9 日の午後6時31分からの2時間で収集したデータをグラフ化したものを図 4 に示す．図 4 はデータ収集を行った 2 時間の間に，接続要求回数の数が 1000 回を超えた IP アドレスの単位時間当たりの接続要求回数の総和である．図 4 より攻撃者は短期間に大量の接続要求を送信してきていることがわかる．表 1 に攻撃者として検出された IP アドレス 583 個のうち上位 80 個と最大接続要求回数を示す．

表 1 1 分当たりの接続要求回数

【SOURCE IP】	【回数】	【SOURCE IP】	【回数】	【SOURCE IP】	【回数】	【SOURCE IP】	【回数】
59.106.15.112	65,097	124.227.213.78	366	222.159.182.51	171	118.2.184.106	95
222.138.229.251	2,039	59.104.254.244	362	118.108.160.29	167	118.2.236.43	92
58.215.93.7	1,280	222.83.142.80	342	195.35.183.60	148	221.78.98.19	92
202.9.117.19	1,208	221.234.227.74	314	202.3.213.3	148	202.92.17.47	92
218.28.160.140	1,024	201.236.101.82	304	81.208.31.222	148	200.248.72.75	87
221.11.6.227	1,024	220.140.114.172	286	122.163.146.178	147	60.238.56.189	86
60.169.3.33	1,024	211.72.161.194	273	219.167.181.87	143	210.203.215.95	85
61.152.110.73	1,024	202.225.112.173	272	133.30.9.193	137	196.211.8.90	76
61.247.108.70	1,024	88.149.192.134	269	62.72.101.154	136	150.214.9.245	69
202.105.113.176	1,020	81.241.231.149	266	83.18.101.134	121	220.221.138.21	68
61.227.1.52	1,003	212.160.173.29	256	194.108.136.72	120	210.138.104.3	66
122.224.34.11	975	80.35.236.230	256	125.197.16.2	115	210.134.110.232	64
122.123.201.169	865	75.142.56.203	252	92.112.149.103	111	87.118.118.98	64
118.170.37.44	764	221.223.200.175	251	118.2.244.85	110	133.5.54.145	63
213.240.230.34	757	83.103.70.170	249	222.150.36.38	110	122.145.15.221	62
219.80.131.74	496	220.111.157.218	214	81.7.76.88	110	61.103.197.154	58
59.113.5.158	486	219.60.118.11	193	125.172.191.41	108	219.113.239.206	58
91.63.120.185	454	145.253.179.228	180	212.165.184.179	106	160.190.234.32	58
124.42.124.87	389	60.53.93.47	174	221.185.147.44	106	121.105.122.65	57
85.64.206.35	369	201.120.51.189	172	202.152.239.214	105	152.71.41.5	56

1分当たり接続要求回数のもっとも多かったIPアドレス(59.106.15.112)(以降，対象ホスト1)と60位IPアドレス(202.152.239.214)(以降，対象ホスト2)が攻撃者であるかどうかについてtcpdumpの出力結果から考察する．対象ホスト1は，SSHのサービスが稼動しているポートに対し，内部ネットワーク内の全てのホストへ接続要求を送信している(図5)．全てのホストに対し接続要求を送信した数分後，反応が返ってきたホストに対し，SSH接続を試みている(図6)．そして，最初の接続要求は1秒間に8，000回以上行われていたが，2度目の接続要求では1秒間に1回から5回の接続しか行われていない．以上のことから対象ホストは最初の1秒間でSSHが稼動しているホストを探索し，探索を終えた数分後から，反応が返ってきたホストに対しSSHのパスワードクラックを試みているものだと考えられる．そのことから，対象ホスト1はSSHサーバを探すscan攻撃を行っていることがわかる．

次に，対象ホスト2に関して検証を行ったところ，内部ネットワーク内のRPC(Remote Procedure Call)サービス(ポート番号 135)が稼動しているホストに対し規則的にアクセスを行っている(図7)．かつ，短時間に大量の接続要求を行っている．よって，対象ホスト2は内部ネットワーク内のRPCが稼動しているホストを探索するためscan攻撃を行っているものと考えられる．

```
19:41:30.117144 59.106.15.112.485 > 133.37.*.0.ssh: S 525425159:525425159(0) win 65535
19:41:30.117272 59.106.15.112.485 > 133.37.*.1.ssh: S 710512716:710512716(0) win 65535
19:41:30.117272 59.106.15.112.485 > 133.37.*.4.ssh: S 767752178:767752178(0) win 65535
19:41:30.117273 59.106.15.112.485 > 133.37.*.3.ssh: S 828132338:828132338(0) win 65535
19:41:30.117273 59.106.15.112.485 > 133.37.*.5.ssh: S 1827065297:1827065297(0) win 65535
19:41:30.117274 59.106.15.112.485 > 133.37.*.2.ssh: S 1774752265:1774752265(0) win 65535
19:41:30.117274 59.106.15.112.485 > 133.37.*.6.ssh: S 1805964142:1805964142(0) win 65535
19:41:30.117393 59.106.15.112.485 > 133.37.*.9.ssh: S 814322052:814322052(0) win 65535
19:41:30.117394 59.106.15.112.485 > 133.37.*.12.ssh: S 1585493686:1585493686(0) win 65535
19:41:30.117394 59.106.15.112.485 > 133.37.*.7.ssh: S 365552873:365552873(0) win 65535
```

図5 1度目の接続要求(対象ホスト1)

```
20:02:36.094760 59.106.15.112.38733 > 133.37.*.1.ssh: S 895820330:895820330(0) win 5840
20:02:36.161596 59.106.15.112.50092 > 133.37.*.10.ssh: S 897823443:897823443(0) win 5840
20:02:37.342908 59.106.15.112.49122 > 133.37.*.11.ssh: S 899891321:899891321(0) win 5840
20:02:37.417865 59.106.15.112.37640 > 133.37.*.12.ssh: S 896835207:896835207(0) win 5840
20:02:37.823253 59.106.15.112.45600 > 133.37.*.13.ssh: S 906677751:906677751(0) win 5840
20:02:37.994279 59.106.15.112.58095 > 133.37.*.14.ssh: S 902930121:902930121(0) win 5840
20:02:38.452887 59.106.15.112.38406 > 133.37.*.15.ssh: S 908666985:908666985(0) win 5840
20:02:38.697871 59.106.15.112.52449 > 133.37.*.16.ssh: S 893711555:893711555(0) win 5840
20:02:38.782195 59.106.15.112.45408 > 133.37.*.17.ssh: S 908098642:908098642(0) win 5840
20:02:39.113377 59.106.15.112.40893 > 133.37.*.18.ssh: S 896413464:896413464(0) win 5840
```

図6 2度目の接続要求(対象ホスト1)

```
20:11:27.250014 202.152.239.214.3022 > 202.253.*.217.135: S 1573414360:1573414360(0) win 64512
20:11:27.250889 202.152.239.214.3032 > 202.253.*.219.135: S 2742962678:2742962678(0) win 64512
20:11:27.251392 202.152.239.214.3018 > 202.253.*.216.135: S 2568969009:2568969009(0) win 64512
20:11:27.252512 202.152.239.214.3027 > 202.253.*.218.135: S 1546503997:1546503997(0) win 64512
20:11:27.257138 202.152.239.214.3014 > 202.253.*.215.135: S 3340654826:3340654826(0) win 64512
20:11:27.645159 202.152.239.214.3036 > 202.253.*.221.135: S 75253613:75253613(0) win 64512
20:11:27.647408 202.152.239.214.3035 > 202.253.*.220.135: S 1677739970:1677739970(0) win 64512
20:11:27.649406 202.152.239.214.3039 > 202.253.*.224.135: S 3571730560:3571730560(0) win 64512
20:11:27.650281 202.152.239.214.3037 > 202.253.*.222.135: S 1437670709:1437670709(0) win 64512
20:11:27.651905 202.152.239.214.3038 > 202.253.*.223.135: S 89364841O:89364841O(0) win 64512
```

図7 接続要求(対象ホスト2)

```
18:31:03.006107 118.2.184.106.3812 > 133.37.*.2.https: S 3019590550:3019590550(0) win 65535
18:31:03.434731 118.2.184.106.3814 > 133.37.*.2.https: S 2114707097:2114707097(0) win 65535
18:31:03.610255 118.2.184.106.3816 > 133.37.*.2.https: S 2373581248:2373581248(0) win 65535
18:31:03.777786 118.2.184.106.3818 > 133.37.*.2.https: S 2939041760:2939041760(0) win 65535
18:31:03.785528 118.2.184.106.3819 > 133.37.*.2.https: S 3375969302:3375969302(0) win 65535
18:31:03.793526 118.2.184.106.3820 > 133.37.*.2.https: S 3845069288:3845069288(0) win 65535
18:31:04.078359 118.2.184.106.3823 > 133.37.*.2.https: S 666357385:666357385(0) win 65535
18:31:04.079357 118.2.184.106.3824 > 133.37.*.2.https: S 517330211:517330211(0) win 65535
18:31:04.082856 118.2.184.106.3825 > 133.37.*.2.https: S 1195580692:1195580692(0) win 65535
18:31:04.082856 118.2.184.106.3826 > 133.37.*.2.https: S 2148828843:2148828843(0) win 65535
```

図 8 接続要求(対象ホスト 3)

接続要求回数が 100 回前後のものは正常の通信であるかを調べるため，対象ホスト 2 と接続要求回数の近い 118.2.184.106(以下，対象ホスト 3)について検証を行った．その結果，送信先 IP アドレスは存在するのだが，同一ホスト・同一ポートに対して短期間に大量に接続要求を行うことはないので，対象ホスト 4 は攻撃を行っているものと考えられる(図 8)．

以上のことから，1 分当たり 100 回以上接続要求が 1 つのホストから来れば，攻撃者であると判断できる．

表 2 に 2008年10月22日午後1時42分から2008年10月24日午後1時42分までの接続要求持続時間に対する接続要求回数を示す．この表は横に各 IP アドレスの 1 分間あたりの接続要求回数の最大値で，縦には，各 IP アドレスが攻撃者であると検知されてから接続要求が 0 になり scan 攻撃などが終了するまでの時間を配置した．この表から 5 分以内に大量に接続要求を行いそれ以降アクセスがないといったものが全体の 57%を占めており，長期的にスキャンを行ってくる攻撃者は少ないということがわかった．

さらに，接続持続時間が長い IP アドレスのトラフィック特性について調べたところ，様々な形があるということがわかった．一例として図 9 を示す．

対象ホストの単位時間当たりの接続要求回数は，攻撃者判定システムと tcpdump を用いて収集したログとで大きな差があった(図 10)．その理由として tcpdump は収集した全パケットのデータをファイルシステムに保存するが，数秒間に大量のパケットを受信した場合，tcpdump がパケットのデータをファイルに保存することが間に合わずパケットをロストしてしまう．一方，本システムは接続要求を行うパケットのみ

表 2 接続要求持続時間と接続要求回数との関係

最大接続要求回数/分

接続要求持続時間(分)	100~199	200~499	500~999	1000~1999	2000~4999	5000~9999	10000~19999	20000~49999	50000~100000	計
0~1	39	34	54	40	7		2	1	7	184
1~5	45	84	75	16	4	3	2	2		231
5~10	4	7	4	3	4	4				26
10~30	8	5	1	9	10	12		1	1	47
30~60	3		1	5	2				1	13
60~120	12	4	2	8	4				1	31
120~360	15	3	7	12	2				1	40
360~720	6	3	11	5		1			3	29
720~1440	19	6	17	7			1		2	52
1440~	32	15	4	14	1	1			3	70
計	183	161	176	120	34	21	5	4	19	723

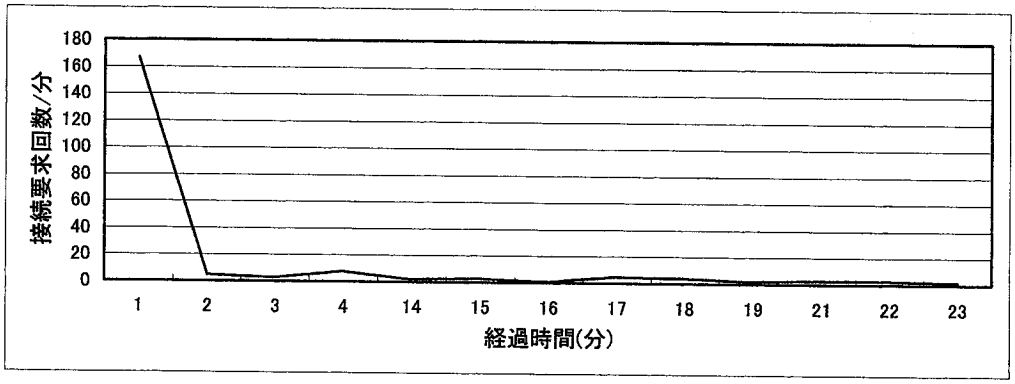

図 9 長期的な scan 攻撃の例

を収集しアクセス回数要求をカウントするだけであるので，本システムが送信元 IP アドレスごとに接続要求回数を計算し保存を行う処理負荷のほうが，tcpdump がファイルシステムにパケットのデータを保存する処理負荷より小さくなる．その結果，本システムのほうが多くのパケットを処理することが可能となるので，本システムと tcpdump との接続要求回数に大きな差ができたものと考えられる．

【SOURCE IP】	【本システム】	【tcpdump】
59.106.15.112	65,097	10,111

図 10 接続要求回数の差

5. まとめと今後の課題

本論文では，攻撃者は短時間で内部ネットワーク内のIPアドレスの情報を得るために短期間に大量のTCP コネクション接続要求を送ってくることから，単位時間当たりのTCPコネクション接続続要求回数から攻撃者を判断できると考え，開発・運用を行った攻撃者判定システムの説明と運用結果について述べた．表 1, 2 より，1 分当たり 100 回以上接続要求が来れば，攻撃者と判断して送信者リストに登録し，最終アクセス時刻から 5 分間経過した IP アドレスを送信者リストから削除が可能であると考えられる．提案したシステムは単位時間当たりのアクセス回数により攻撃者を判断しているので，帯域幅を小さく絞ったスロースキャンには対応していない．スロースキャンの検出は文献[5]で述べている．

長期的なログを収集し，攻撃者であると判断し，抑制できる保持時間，単位時間当たりの接続要求回数のしきい値とを調べることが今後の課題である．さらに，提案したシステムは攻撃者であると判断されたIP アドレスに対して LAN スイッチのアクセス制御リストを用いた遮断を行うシステムを構築し追加することでscan 攻撃抑制システムとしたい[6]．また，パケットが大量に届いたときパケットをロストしてしまう可能性があるので，パケットをロストしたときの対策，またパケットをロストしないためにシステムの構築を行うといったことを今後の課題である．

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] フレッツ・セーフティ, http://flets.com /customer/tec/safety/index.html

[3] psad, http://www.cipherdyne.org/psad/

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

[5] 兒玉清幸，藤原健志，大塚賢治，吉田和幸，“長期的スキャニングを対象としたスキャン攻撃検知システムの評価”，情報処理学会研究報告，

2008-IOT-2, pp7-12, 2008/7/24
[6] 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, 2008/9/24
[7] 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
[8] 南浩一, 吉田和幸, “throttlingを用いたscan攻撃抑制システム”, 分散システム/インターネット運用技術シンポジウム2006年度論文集 情報処理学会シンポジウムシリーズ Vol.2006 No13, pp.43-47, 2006
[9] 南浩一, 吉田和幸, “throttlingによる攻撃抑制の効果について”, 情報処理学会研究報告 (2007-DPS-130/2007-CSEC-36), pp.381-386, 2007