ランダムで低速なポートスキャンの検知についての検討

武仲正彦 †　　鳥居悟 †　　清水聡 ‡

† 株式会社富士通研究所
211-8588 神奈川県川崎市中原区上小田中 4-1-1
‡ 株式会社富士通ソーシャルサイエンスラボラトリ
211-0063 川崎市中原区小杉町 1-403 武蔵小杉タワープレイス

あらまし 近年のサイバ攻撃は, その手口が巧妙になってきている. 例えばポートスキャンにしても, 従来のように短時間にシーケンシャルにスキャンを行うのではなく, 一見ランダムなポートを長期間かけてスキャンすることで, 侵入検知システムを逃れるような方式が行われ始めている. 本稿では, 実際のネットワークログからこのようなスキャンが検出されたことを示し, ランダムで低速なポートスキャンを統計的手法で検出できるかどうかの検討を行う.

Study on Detection for Randomly Slow Port Scanning

Masahiko Takenaka†　　Satoru Torii†　　Satoru Shimizu‡

†FUJITSU LABORATORIES LTD.
2-1-1 Kamikodanaka, Nakahara-ku, Kawasaki 211-8588, Japan
‡FUJITSU SOCIAL SCIENCE LABORATORY LIMITED
Musashi-Kosugi Tower Place 1-403, Kosugi-machi, Nakahara-ku, Kawasaki, 211-0063, Japan

Abstract Recently, the way of cyber-attack has become more tactical and tricky. For example, traditional port-scanning sequentially scans ports briefly. But, current ways randomly scan ports in a slow pace, then, these scans cannot be detected by IDS (Intrusion Detection System). In this paper, we show the fact that we have detect such scans from access-logs of a real-network. And, we study and discuss detection method according to only statistics of network-accesses.

1　はじめに

近年, サイバ攻撃が激化し, 政府機関や防衛産業企業等が攻撃を受け, 不正侵入や情報漏えいの被害が発生している. 特に APT 攻撃と呼ばれる, 特定の組織や個人を標的にした, 複数の攻撃手法を組み合わせ執拗かつ継続的な攻撃は, その手口が巧妙で防御が困難となっている.

最近公表された財務省への攻撃では, 数年前から攻撃が行われていた上に, 侵入したマルウェアの一部は最新の検査ツールでも検出できず, 通信ログの解析による疑わしい通信の検出から初めて侵入を検知できたと言われている [1].

本研究は, 富士通が管理しているネットワークの 4 か月間のログから, 一見ランダムで非常に低速なポートスキャンと思われる通信が, 手作業により抽出されたことを契機としている. ポートスキャンは, サーバ内で動作しているアプリケーションソフトや OS の種類, 侵入口となりうる脆弱なポートがないかどうか調べる行為で, 不正侵入の前段階に実施されることが多い. そのため, この段階での攻撃通信の発見は, 対策にとって非常に重要であると言える.

従来のポートスキャンでは, 短期間で脆弱なポートを発見することを目的としていたため,

短時間でシーケンシャルにスキャンを行うものが多く, 組織ネットワークの出入り口における侵入検知システム (IDS/IPS) で, 時間当たりの通信数等の監視により検出が可能であった. これに対し, 今回抽出されたような, 一見ランダムなポートに対する長期間のスキャンは, 従来の IDS/IPS では, 検知効果が期待できない. こうした攻撃を検出するためには, 長期間収集したログから抽出するしかなく, 膨大なログから如何に目的の通信を抽出するかがポイントとなってきている.

本稿では, 2 章で, 今回観測された通信に対する手作業による抽出結果より, 実際にこのような非常に低速なポートスキャンが実施されていることを示し, その特徴について述べる. そして, 一般的な IDS でのポートスキャン検出について述べ, それを回避するようなスキャンツールを示すことで, 非常に低速なポートスキャンの可能性について議論する. 3 章では, 低速なポートスキャンを検知するために, 従来のような時間比率を用いず, 統計量のみにより検知を行う方法について検討を行い, 4 章では, それらの手法を実際のログに対して適用し, その検出能力についてディスカッションを行い, 5 章で, まとめと今後の課題について述べる.

2 ランダムで低速なポートスキャンの発見

最近のサーバ攻撃の激化を受けて, 富士通が管理しているネットワークでも, 昨年 12 月から 4 か月間のログの調査を実際した. その結果, 非常に低速であるが, 多数のポートに対してポートスキャンと疑われる通信を行っている, 特定のサーバがあることが判明した.

2.1 観測結果

抽出作業では, ポートスキャンと判断するために, 疑わしい通信群に対し, 横軸を時間, 縦軸をポート番号のグラフにプロットして, 目視で何らかの特徴があるものを抽出した. 抽出した 2 種類のスキャンのグラフを図 1,2 に示す.

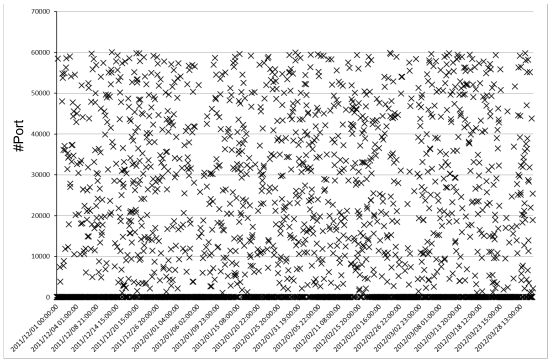

図 1: 検出したスローポートスキャン#1

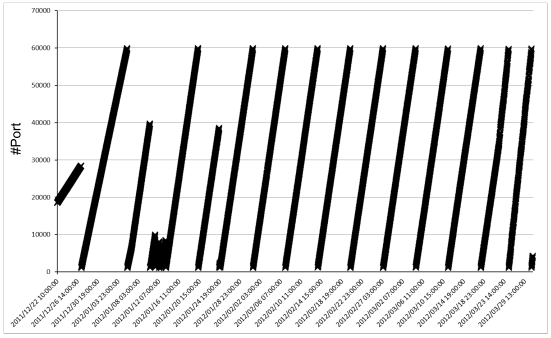

図 2: 検出したスローポートスキャン#2

手動抽出したスキャンの特徴は以下の通り.

- アクセス頻度が少ない
  - #1：2362 回/4 か月 (平均 0.8 回/時)
  - #2：9361 回/4 か月 (平均 3.2 回/時)
- アクセスポートはシーケンシャルでない
  - #1：ランダム（Port：0 番を除く）
  - #2：飛び飛び
- アクセス先はオーバーポート (1024〜65534)
- 頻度はほぼ 1 回限り (#1 の Port：0 番を除く)

これが手動で抽出検出できたのは, グラフ化した時に特徴があったためである. #1 は Port:0 を複数回スキャンするという, 不審な挙動によるり検出した[1]. #2 は時間に対するスキャンポート番号が直線に乗っていることから検出した[2].

[1]Port:0 をスキャンするのは, 明らかに無駄な上に IDS 等での検出の可能性を上げるだけなので, スキャナツール/設定のバグではないかと想像している.

[2]#2 はグラフ上では直線に見えるが, 実際にスキャンされたポートは飛び飛びの値を取っている.

これらの検出には, グラフ化と目視確認が必要であり, ある程度スキャンが実施された後でなければ検出することができない上に, 大きな手間が必要である. また, グラフに特徴が無ければその検出は困難となる. 例えば#1 の Port:0 のスキャンが無いような場合, グラフには特徴が表れず, 目視による検出は困難である. そのため, そのようなスキャンは, まだこの 4 か月間のログ内に含まれている可能性があり, このような非常に低速のスキャンを効率的に検知する手法が求められる.

2.2 IDS でのポートスキャン検知

まずは, 従来のネットワーク型 IDS でのポートスキャン検知について調査を行った. 代表的なオープンソースのネットワーク型 IDS である snort [2] の低速ポートスキャンの検出性能について述べる.

ポートスキャンの時間間隔 (time window) に関するパラメタとしては 'sense_level' が用意されおり, 'low', 'medium', 'high' の 3 種類のレベルが指定可能である. 'sense_level' が 'Low' の場合 60 秒間, 'Mid' なら 90 秒間, 'High' なら 600 秒間に検知したポートアクセス数が閾値を超えたり, 特定のパターンを示した場合に, ポートスキャンを受けたと判断する. この時間間隔より低速なポートスキャンの場合, snort では検出が困難となる. この時間間隔を拡大すると検証しなければならない通信量が膨大になる事は明白であり, リアルタイム検出を目的とするネットワーク型 IDS/IPS では対応することは困難である.

2.3 ポートスキャニングツール

一方, ポートスキャニングツールにとっては, 非常に低速なポートスキャンは, 調査効率が低下するだけで, 技術的な課題は存在しない. 例えば, オープンソースの代表的なポートスキャニングツールである nmap [3] でも, デフォルトで非常に低速なポートスキャンが実施可能である.

nmap のスキャンテンプレートを 'Paranoid' (-T0) とすれば, スキャン間隔は最低 5 分となる[3]. スキャン間隔をより低速にする場合は, '–scan_delay' オプションで直接間隔を指定ることで可能となる.

さらに, nmap ではデフォルトでスキャンするポート順序はランダムであり, 手作業によるポートスキャン抽出作業において目視による判別がより困難となる.

2.4 検出の課題

2.2 節で述べたように, 非常に低速なポートスキャンはネットワーク型 IDS/IPS が用いていような, 単位時間当たりの通信数といった手法での検出は困難である.

一方, 2.1 節のように特徴的なグラフ形状を持つものを抽出する場合は, 図 1 の Port:0 スキャンが無いようなものは検出することが困難である. 実際, 2.3 節で述べたように, nmap ではスキャンするポートをランダムに選択することができる.

単純にログを観測して, スキャンされたポート数が多ければ検出するという方法も考えられる. しかしこの場合, 「多い」という閾値をどう判別するか, 意図的なポートスキャンかどうかをどう判別するかが課題となる.

そこで本稿では, ポートスキャンでは「スキャンするポート番号が重複しない」という特徴に注目し, 検出手法を検討する.

3 検出手法の検討

本章では, 非常に低速のスキャンを効率的に検知する手法について検討する.

ポートスキャンは, 対象のポートを網羅的にスキャンすることが目的である. 非常に低速にスキャンを行うためには, ただでさえ全ポートのスキャンに時間がかかるので, スキャン自体を効率的に行わなければならない. 効率的なポートスキャンを行うためには,「スキャンするポート番号は極力重複させない」ということは重要な要件である. この特徴を直接抽出できれば, 低

[3]マニュアルの 'Paranoid' の説明には, 明確に IDS を回避するオプションと記述されている.

速なポートスキャンの効率的な検出が可能であると考えた.

3.1 通常通信との識別

重複がないことで, 非常に低速なポートスキャンを検出には, スキャンと通常通信を識別する必要がある. 一般に通常通信には様々な通信パターンがあり, 単純にモデル化することは困難である. しかし, 使用するポートに着目すれば, 通常通信では使用するポートに偏りがあるのに対し, 重複がないスキャンは, 全ポートに対して「一様な通信」を行うと言える. そこで本稿では, 全ポートに対して「一様な通信」を識別することを考える.

識別の上で「一様な通信」を「一様すぎる通信」と「ランダムなポートへの通信」に分類する. 「一様すぎる通信」は, シーケンシャルに全ポートをスキャンしたり, そのスキャン順序をランダムに決定する等, スキャンする全てのポートが一度しか現れないものと定義する. 例えば, 2.3 節で述べた nmap によるスキャンがこれに相当する. 一方, 「ランダムなポートへの通信」は, 一様乱数でスキャンするポート番号を定めた場合と定義する. この場合は, ある程度使用ポートの重複が現れるが, 全ポートへの「一様な通信」を目的としていると言える. また, スキャン順序をランダムに決定するような「一様すぎる通信」を複数回実施した場合なども, 同じ傾向となる.

「一様すぎる通信」と「ランダムなポートへの通信」を比較すると, 「一様すぎる通信」の方が効率的にスキャン可能である. 特に, 非常に低速なポートスキャンにおいて, 全てのポートを網羅的にスキャンしたい場合は, 「一様すぎる通信」を用いるのが最も効率的である. そこで本稿では, まず「一様すぎる通信」と「ランダムなポートへの通信」の識別を行い, そののち「ランダムなポートへの通信」の統計量による識別を行うことで, 通常通信とスキャンとの識別を検討する.

3.2 重複が無いことに着目した検出手法

「一様すぎる通信」ではないことは, 観測した通信の数とそのポートの種類が一致するかどうかで判断可能であり, 容易である. 一方「一様すぎる通信」と観測した通信だけで判断するのは, 閾値の設定が容易でない. そこでまず, 「一様すぎる通信」と「ランダムなポートへの通信」の識別を行う. 一様乱数で繰り返しポートを決定した場合, 繰り返し回数が増加するに従い, 「ポートが重複しない」確率が減少する. その確率が有意なほど小さくなることで「一様すぎる通信」と「ランダムなポートへの通信」を識別する.

スキャンするポートの総数を n, スキャンが測定されたポートの個数を k とすると, k 個まで一度もポートが「重複しない」確率 P_k は次のような漸化式で表現できる.

$$P_k = \frac{n-k+1}{n} P_{k-1} \qquad (P_0 = 1)$$

これを展開すると, 次のような式となる.

$$P_k = \frac{n!}{n^k (n-k)!}$$

全ポート (1 〜65535) の場合, 各棄却確率での k の値は表 1 のようになる.

表 1 より, 「ポートが重複しない」通信が連続 626 個測定できれば 95%の確率で, 連続 1097 個測定できれば 99.99%の確率で, 「ランダムなポートへの通信」ではなく「一様すぎる通信」であると識別可能である.

3.3 χ^2 値による検出手法

次に, 「ランダムなポートへの通信」の識別を検討する. 「ランダムなポートへの通信」は, 乱数の一様性検証が利用可能である. 一般に一様乱数かどうかの識別には χ^2 検定が利用される. 観測された通信ポートの χ^2 値を計算し, 極端に上方, 下方に振れなければ一様乱数とみなすことができる. 上方 (数値が大きい方) に振れた場合は, 一様乱数より偏っていると判断でき, 「ランダムなポートへの通信」ではなく, 何らかの通常通信の可能性が高まる. また, 下方 (数値が小さい方) に振れた場合は, 3.2 節の「一様すぎる」通信の可能性が高まる.

表 1: 値域 (1〜65535) における乱数の重なり確率による棄却値

下方棄却	5%	1%	0.5%	0.1%	0.05%	0.01%
k	626	776	832	950	997	1097

表 2: 値域 (1〜65535) の χ^2値の上方, 下方棄却値

上方棄却	5%	1%	0.5%	0.1%	0.05%	0.01%
χ^2 値	66131.6	66380.1	66471.3	66659.5	66732.8	66890.0
下方棄却	5%	1%	0.5%	0.1%	0.05%	0.01%
χ^2 値	64940.6	64695.7	64606.2	64421.9	64350.3	64197.1
k	596	841	930	1115	1186	1339

χ^2 値による識別方法を示す．スキャンするポート個数を n, スキャンが測定されたポートの個数を k, 測定された各ポートのスキャン回数を x_i とすると, 測定された通信の χ^2 値は以下の式で表現できる．

$$\chi^2 = \sum^{k} \frac{(x_i - \frac{k}{n})^2}{\frac{k}{n}} = \sum^{k} \frac{(nx_i - k)^2}{kn}$$

表 2 に, ポートが 1〜65535 ($n = 65535$) の場合の上方下方棄却値を示す．本稿では Excel[4] の関数 CHIINV を用いてい棄却値を計算している．計算方法は CHIINV(確率, 自由度)($= n - 1$)) である．

ここで, 比較のために, 3.2 節と同様に, スキャンポートの重複が全く無い場合について, χ^2 検定で初めて検出される個数 k を算出する．$k < n$ の場合, x_i のうち k 個が 1 で $n - k$ 個は 0 であることから, 以下のように変形できる．

$$\chi^2 = \frac{k(n-k)^2 + (n-k)k^2}{kn} = n - k$$

以上のことから, 重複が全くない場合の χ^2 値は測定されたポートの個数の一次関数で表現できる．

図 3 に 0〜255 までの一様乱数と, それを重複が無いように選択した場合の χ^2 値の実験結果を示す．一様乱数の場合 χ^2 値は一定の範囲内で推移するのに対し, 重複がないようにランダムに決定した場合の χ^2 値は, 理論通り単調減少することが判る．そのため, 重複がないようなポートスキャンは, 測定数が少ないうちから「ランダムなポートへの通信」との区別が可能となるため, より検出しやすいと言える．

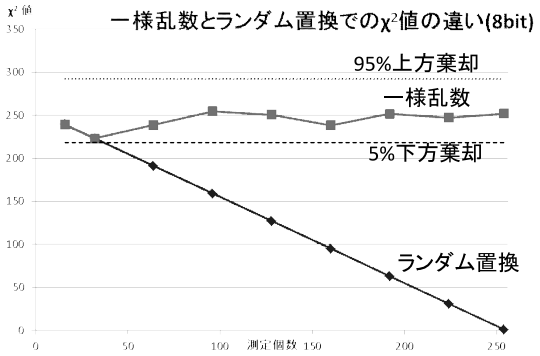

図 3: 一様乱数とランダム置換でポート番号を決定した場合の χ^2 値の変化

4 検証結果

検討した方式について, 2 章で述べた 4 か月間のログで検証を行った．

4.1 「一様すぎる通信」の抽出

検証の結果「一様すぎる」通信は発見できなかった．

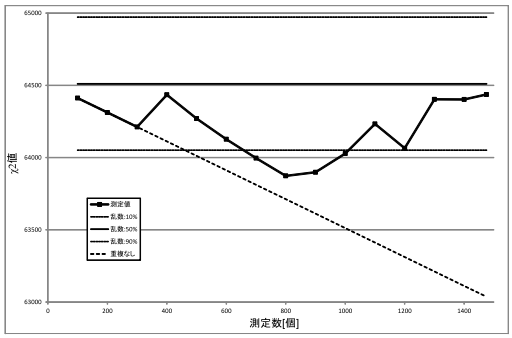

図 4: 測定数に対する χ^2 値の変化

4.2 「ランダムなポートへの通信」の抽出

次に,「ランダムなポートへの通信」の抽出を行ったところ, 2.1 節の#1 と類似した通信が複数検出された. 検出された通信の測定数に対する χ^2 値の変化を図 4 に示す. この通信は 10 スキャン/日程度で, 対象ポートはオーバポート (1024〜60000)[4]であった.

他の検出通信についても検証したところ, 攻撃元のサーバ (IP アドレス) や通信頻度は異なるものの, この通信とほぼ同じ特徴を持つことが判明した.

4.3 考察

「一様すぎる通信」は, 最も効率的なポートスキャンが可能であり, nmap のデフォルト機能で実施可能であるのに, 発見できなかったことは意外である. この手法はスキャンの途中でツールを再起動させたるだけで検知できなくなるためかもしれない. 長期間の動作が必要な非常に低速なスキャンでは, 攻撃者はツールの再起動を実施している可能性がある. 例えば 4 では, 300 スキャンと 700 スキャンあたりで急激に χ^2 値が上昇している. 通信回数が 10 スキャン/日程度であることを考慮すると, ツール自体は「一様すぎる通信」を行ってるが, 1 か月に 1 度程度ツールを再起動している可能性がある.

発見した「ランダムなポートへの通信」については, 攻撃元のサーバ (IP アドレス) や通信頻度は異なるものの, Port:60000 以下のオーバポートにランダムな通信を行うという特徴を持つものが複数見つかっている. この不自然な特徴を共有するとこうことは, 明らかに同じツールを用いたものであると考えられ, ランダムで非常に低速なポートスキャンを実施する何らかのツールが出回っている可能性がある.

[4]通常オーバポートは〜65535 であるが, この攻撃者は何故か Port:60000 までの通信を行っている.

5 まとめと今後の課題

本稿では, 非常に低速でランダムなポートスキャンが実際に行われていることを報告し, そういったスキャンを効率的に識別を目的に, ポートの重複がないことに着目した検出手法の検討を行った.

これらの手法を用いて, ログを検証したところ,「ランダムなポートへの通信」によるポートスキャンと思われるものが複数検出された. また, それらのポートスキャン通信は不自然な特徴を共有しており, 何らかの同じツールによるスキャンであると考えられる.

今後は, 他のネットワークにおけるログにも適用を行い, 非常に低速でランダムなポートスキャンの実態を収集し,「一様すぎる通信」や「ランダムなポートへの通信」といった特徴でポートスキャンを効率的に検出できないかの検証を行う.

参考文献

[1] 財務省,「財務省におけるウイルス感染事案について,」http://www.mof.go.jp/about_mof/other/other/press_20120720.html

[2] Snort.org, “Snort: open source network intrusion prevention and detection system,” http://www.snort.org/

[3] Nmap.org, “Nmap Security Scanner” http://nmap.org/

[4] Microsoft Cop, “Excel 2010,” http://office.microsoft.com/excel/