3zc-03 順序データからのマイニングとその不正侵入検出への応用*

伊藤大介† 大和田勇人† 溝口文雄†

東京理科大学 理工学部‡

1 はじめに

本論文では外部ホストからの接続要求に関するログデータから、そのイベントの順序構造に着目したマイニング法とそのアルゴリズムを示し、不正侵入検出に役立つルールを抽出する。

不正侵入には数多くの手法が確認されているが、本研究では順序データからのマイニングに最適であると考えられるポートスキャンに着目する。ポートスキャンは外部から侵入する際の下調べとして非常によく用いられる手段であり、この形跡を早期に発見することによって次の不正行為に対する備えが可能になる。

2 ポートスキャン

Full-Connection 型ポートスキャンは最も一般的に用いられ、一度に複数のポートやホストの状態を調べることができ効率的であるため管理者が管轄している機器の検査にもよく用いられる。また TCP の Full-Connection を確立する方式をとるため、正常な外部からの接続方法であるという認識がなされ、スキャニングに対する制限は受けず、メッセージに異常も残らない。このような侵入の兆候を発見するには今回提案する順序データによるマイニングが有効である。

そしてそのログデータは、図1に見られるように短い時間間隔で起こるシリアルエピソード [1] である。そこには順序関係が存在し、一定の時間内で決められた数のイベントをもっている。しかし侵入を試みようとしている者すべてが、このように目で見てあきらかなポートスキャンの兆候を残すわけではない。侵入者は空いているポートを知るのが目的なのだから、一回に調査するポートの

$$in.ftpd \xrightarrow[21]{} in.telnetd \xrightarrow[1(s)]{} in.timed \xrightarrow[23]{} gn \xrightarrow[19]{} in.fingerd \xrightarrow[37]{} ipop2d \xrightarrow[47]{} ipop3d \xrightarrow[70]{} imapd \xrightarrow[13]{} in.rlogind \xrightarrow[143]{} in.rshd \xrightarrow[561]{} \xrightarrow[79]{} \xrightarrow[41]{} \xrightarrow[109]{} \xrightarrow[2]{} \xrightarrow[110]{} \xrightarrow[42]{} \xrightarrow[513]{} \xrightarrow[0]{} \xrightarrow[514]{}$$

図1 ポートスキャンエピソード

※ イベントの下の数字はポート番号、
→の下は要した時間 (秒) とする

数を 20 ずつに分け、それぞれの間に十分な時間間隔をとれば、他ホスト、もしくはスキャニングを行ったそのホストからの接続情報が混ざってしまい、その検出は困難なものとなる。そこで本研究のアルゴリズムは、発見するエピソードの断片部分からの順序データの一致でも他ホストからのポートスキャンの検出を可能にするルールを抽出する。

3 アルゴリズム

今回のマイニングにはまず順序データを参照するための情報が必要である。どのポートが空いているかはそれぞれのホストによって異なるため、事前に自ホストの空いているポートを調べておき、その情報を参照するファイルに格納しておいて検出の際に用いる。参照ファイルには、図1のようなイベントの順序関係と各イベントごとにおける時間軸の関係を記述する。

アルゴリズムは次のようである。C_iを侵入行為に対する候補エピソード、E_oを要素別のイベントタイプのクラスとし、E_oに含まれる e というイベントの集合についてマイニングを行なう。その結果として参照ファイルと一致した部分を返す。

1. $C_1 := \{\{e\} \mid e \in E_o\}$;
2. $i := 1$;
3. **while** $C_i \neq \phi$ **do**

*A mining approach using sequential data to intrusion detection

†Daisuke Ito,Hyato Ohwada,Fumio MIZOGUCHI

‡Faculty of Sci. and Tech., Science University of Tokyo

4. 順序性と時間軸との関連において、
参照ファイルと一致した内容を L_i とする
5. i := i + 1;
6. **od**;
7. **for** all i, output L_i

このアルゴリズムの特徴は次のようにまとめられる。

1、Time Window による分割とその方法

本アルゴリズムでは Time Window を用いて、そのトランザクション内の順序データと参照ファイルとの照合を行う。つまり一般の侵入検出のようなログデータの一行単位、一つのイベント単位での不正な振る舞いの検出ではなく、そのひとかたまりのトランザクションにおける順序データとしてマイニングを行い、ひとたび異常が発見された場合は、そのトランザクションの番号を返すという形をとっている。

そしてその分割方法として、シーケンスの連続性の度合いを知るために一定の時間間隔で区切っていくのではなく、前後のイベントの時間間隔で区切ってそれぞれのイベントの相関の有無を調べる。こうすることによってセキュリティに関連するシリアルエピソードの発見が容易になる。

2、複数の Window width の使用

1つのポートをスキャンするには平均1～2秒の時間がかかり、各イベントの間には空いていないポートを調べている時間も多い。しかし空いているポートのポート番号が隣接しているような場合は極めて短い時間で2つのイベントは発生する。

これらから Window width に変化をもたせることにより、各イベントとポートスキャンエピソードとの時間軸による関連性の比較が可能になる。ポート番号が近いにも関わらず、そこに長い時間間隔があるような場合は、順序性において一致が見られても、それは偶然であると判断できる。逆に順序性の一致がごく断片的なものであっても時間軸による一致が見られた場合、それはポートスキャンの兆候である。

4 方法

実験は RedHatLinux5.2 の Secure ログを使用して、参照ファイルに一致した順序データと実際にポートスキャンを実行した部分との照合をすることにより評価を行った。Window width は 10、30、60、120、240、480、960（単位秒）の7つを使用し、図1のようなポートスキャンエピソードをもつホストを実験対象とした。

各順序データにおける結果の評価基準は、以下の式で算出されるマッチ率で求める。

マッチ率(%) =実際にポートスキャンを行った部分の合計／参照ファイルと一致した部分の合計

5 実験結果

実験の結果、以下のようなルールが発見された。

表1 結果 (Ww は Window width)

No.	Ww	イベント	マッチ率
1	10	in.ftpd → in.telnetd	88.9
2	10	ipop2d → ipop3d	100
3	10	in.rlogind2 → in.rshd	100
4	30	in.ftpd → in.telnetd → in.timed	100
5	30	gn → in.fingerd	100
6	60	in.ftpd から imapd	100
7	960	in.ftpd から in.rshd	100

表1より断片的なポートスキャンの検出に有用であると考えられるルールは、No. 2、3、4、5の4つがあげられる (ただし4に関しては、ftp と telnet の間隔は 10 秒以内がさらに望ましい)。すなわち一般のユーザはごく短い時間にこのような振る舞いをしないということがいえる。このような場合は断片的な順序データであろうとも、まずスキャニングを受けていると考えてよい。

それに対し1の ftp から telnet という行動パターンは、たとえ 10 秒という短い間隔であっても、ユーザの振る舞いとして十分に考えられる。

6 おわりに

これらから同一ホストからの、ポート番号が昇順の順序データにおける Full-Connection 型ポートスキャンの形跡の検出に有用なルールを得ることができた。この研究で断片的なポートスキャンの検出が可能となるが、検出対象が一つのホストからの接続に限られてしまうのが今後の課題である。

参考文献

[1] Heikki Manila, Hannu Toivonen, A.Inkeri Verkamo: Discovering frequent episode in sequences, KDD-95, 1995.