マルチメディア通信と分散処理　97－37
コンピュータセキュリティ　8－37
（2000．3．22）

パケットのデータサイズ・時刻系列を使った コネクション・チェインの発見

依田邦和　江藤博明

日本アイ・ビー・エム (株) 東京基礎研究所

インターネットでは侵入者達は通常踏み台ホストと呼ばれる複数のコンピュータシステムへ連鎖状にログインした後にターゲットへ侵入するため，侵入の大元を突き止めるのは非常に困難である．本論文では侵入者が telnet や rlogin などのインタラクティブなアプリケーションを使って踏み台ホストを長時間アクセスしたとき，連鎖の上流の踏み台ホストを見つける問題を扱う．インターネット上の様々な地点でパケットの時刻とヘッダを記録したデータが得られるとして，これらを利用する．我々はコネクションのパケット系列について“偏差”を定め，ある踏み台での侵入者の用いたコネクションのパケット系列に対する様々なトラフィック点で記録した大量のコネクションのパケット系列の偏差をそれぞれ計算する．この値が小さければそれは同一チェイン上にある可能性が高い．一方，全く関係の無いコネクションについてはこの値は大きいということを実験データで示す．

Finding A Connection Chain Using Pakcet Sequences of Data Sizes and Time Stamps

Kunikazu Yoda　Hiroaki Etoh

Tokyo Research Laboratory, IBM Japan, Ltd.

Intruders on the Internet usually login through a chain of multiple computer systems called step-through hosts before breaking into their targets, which makes tracing the origin really difficult. In this paper, we address the problem of finding step-through hosts in the upper stream of the chain when an intruder used telnet or rlogin to access these hosts for a long time. We assume that time stamps and header information of packets are logged and available at various traffic points in the Internet. We define 'deviation' on packet sequences of connections, and compute deviations of packet sequences recorded at the traffic points from the packet sequence of the intruder's. If a deviation is small, the connection of that packet sequence is likely to be in the same connection chain of the intruder's. We also show some experimental results that the deviation of completely different packet sequence from a given sequence is large.

1 はじめに

近年コンピュータシステムへの不正侵入が増加しており，より多くの商業的な活動やサービスがインターネット上で行われる将来はますます増加することが懸念される．コンピュータネットワークにおける不正侵入の特徴の一つに，侵入が発覚したとしても侵入者（の使用するコンピュータ）を突き止めるのが非常に困難であることが挙げられる．

侵入者は通常踏み台と呼ばれる不正にログインできるコンピュータを予めインターネット上に数台確保している．インターネット上にはセキュリティの甘いコンピュータが数多く存在し，それらを自動的に探すツールが広く出回っていて操作も簡単なため，踏み台に利用できるコンピュータは容易に見つかる．彼等は自分達のコンピュータから直接ターゲットのコンピュータには侵入せず，まず踏み台のコンピュータにログインして，そこからさらに別の踏み台にログインして，これを何度か繰り返してから最終的なターゲットに侵入する．

彼等は侵入した踏み台やターゲット上のコンピュータのログをたいてい消去してしまうが，もしログが残っていたとしても，それから分かるのは侵入を受けた一つ前の踏み台のコンピュータのアドレスだけである．したがって侵入の大元を突き止めるためには，もし全てのログが残っていても，この踏み台の連鎖を一つ一つ辿って調べなければならない([4] など参照)．踏み台にされるのは様々な国のずさんな管理下のコンピュータであるかもしれず，それらの管理者達に連絡をとって，順に調査していくのは大変手間と時間のかかる作業であり，大抵は途中の踏み台でログが全て消されているためそれ以上連鎖を遡れなくなってしまう．

ユーザの手元のコンピュータ H_0 から中間ホスト $H_1, H_2, \ldots$ を経てターゲット H_n に到る連鎖的にログインされたコンピュータの列のそれぞれの間に張られたTCPコネクションの列 $C_1, C_2, \ldots, C_n$ を“コネクション・チェイン”（以下では略して“チェイン”とも記す）と呼ぶ（図1参照）．このようなチェインを追跡する方法に関する研究はこれまでに [1, 2, 3, 6] などが知られている。基本的なアイディアは

1. 管理ネットワーク下の全てのログイン情報を一箇所に集約して常に状況を把握するタイプ
2. 各ホストがログインの際に，チェインの上流のホストの状態を確認するタイプ
3. ネットワーク内の多くの場所でトラフィック・データを記録しておき，侵入を受けた後に，侵入に使われたトラフィックと関連するトラフィックを探すタイプ
4. 侵入を受けている最中に能動的に追跡するタイプ

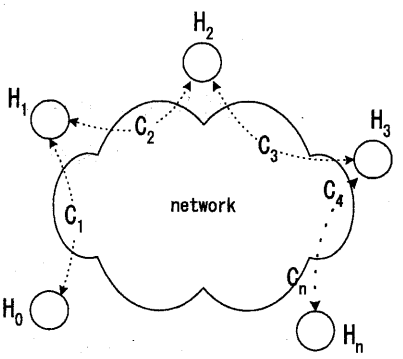

図 1: コネクション・チェイン

などである．我々の手法は3のタイプであり，ネットワークを流れるパケットの情報を記録しておき，侵入者のものと似たパケット系列をネットワーク内の至るところで記録されたパケットデータの中から見つけ出す．

本論文はまず第2節で関連研究の解説をする．次に第3節で同一チェイン上のコネクションを見つけ出す我々の手法を説明し，第4節で実験結果を示す．

2 関連研究

これまでに研究されたコネクション・チェインを追跡するシステムについて以下に紹介する．

DIDS(Distributed Intrusion Detection System)[2] は，管理下のネットワーク内で起こる全TCPコネクションとログインを監視して，ユーザの移動や現状を常に把握する手法である．各ホストにはそれぞれホストモニターが動いていて，監査情報を収集して中央のDIDSディレクターへ

送り，そこでネットワーク状況が集中管理される．

CIS(Caller Identification System)[1] はログインの際に通信元を確認する手法である．N 番目のホストにログインする際，このホストは1から $N-1$ 番目のそれぞれのホストに各自の上流ホストのリストを問い合わせ，結果に食い違いがなければログインを許可する．管理下にあるホストには CIS を導入することを前提とする．

Caller ID[6] はアメリカ空軍により報告されたシステムで，侵入されている踏み台ホストの連鎖を逆向きに侵入者と同様の方法で“侵入”していく手法である．探索側は，すでに侵入された事実から同様に侵入できると主張しているが，実際は困難であったり侵入者によるセキュリティホールの修正などで不可能となる場合もある．また第三者のコンピュータに侵入することはあらたな犯罪とも考えられる．

不正アクセス追跡の手法は，追跡を行うためのコンポーネントをどこに設置するかによって，「ホストベース」と「ネットワークベース」に分類できる．ホストベースの手法は追跡をするためのコンポーネントを各ホストに設置するのに対して，ネットワークベースの手法は追跡をするためのコンポーネントをネットワークのインフラに設置する．DIDS[2], CIS[1], Tsutsui[5], 平田ら [7] の方法はホストベースの手法である．

ホストベースの手法の問題は，チェインの途中のある踏み台がそのシステムを導入していなかった場合，それより上流の追跡が不可能という点である．インターネットではある特定のホストベースのシステムを全ての管理ドメインのホストが採用するということは考えにくい．また，管理下のホストであったとしてもそのホストが侵入されて追跡に関わるプログラムが書き換えられていることも考えられる．したがってインターネット環境ではホストベースのシステムは現実的ではない．

次にネットワークベースの手法を紹介する．Thumbprinting[3] は通信データの内容のみに着目し，データの特徴量 (通信文字種の分布) は個々のセッション（侵入行為）ごとにユニークであり，チェインのどのコネクションにおいてもほとんど同じであるという仮定に基づく．ネットワーク上のできるだけ多数の地点で，各セッションに対して一定時間間隔毎に区切ってこの特徴量をそれぞれ計算して保管しておく．もし侵入が発覚した場合，侵入に使われたセッションの特徴量と近い特徴量を持つものをネットワーク上の多数の地点で探すことでチェイン上にあったホストを発見する．

ネットワークベースの手法の利点は，インターネットで部分的にでもこの手法が取り入れられていれば，それが役に立つ事である．つまりチェインが一つずつ順番に見つかるわけではなく，この手法が取り入れられている範囲内にあればその部分のチェインが見つかるのである．また通信内容を記録する装置は，自分からは全く送信を行わない完全に受動的な機能のみに限定すれば良く，このような装置は侵入を受けにくい．

3 同一チェイン上のコネクション発見

3.1 コネクション・チェイン

コンピュータ H_0 を使用するユーザが他のコンピュータ H_1 へネットワーク越しにログインしたとする．するとこの2つのコンピュータ間には TCP のコネクション C_1 が確立される．そのユーザが H_1 からコンピュータ H_2 へログインし，その後同様にして $H_3, \ldots, H_n$ へ連続してログインしたとすると，それぞれに対して異なるコネクション $C_2, C_3, \ldots, C_n$ が確立される．我々はこのコネクションの列 $C_1, C_2, \ldots, C_n$ を“コネクション・チェイン”と呼ぶ．

TCP のコネクションは (始点 IP アドレス，終点 IP アドレス，始点ポート番号，終点ポート番号) の四組で一意に決まるので，パケットのヘッダ (IP,TCP ヘッダ) を見ることでそれがどのコネクションに属するのか判別できる．

3.2 問題

我々が扱うのは，「コネクション・チェイン $C_1, C_2, \ldots, C_n$ のどれか一つのコネクション C_k に属するパケット系列が得られたとき，それより上流のコネクション C_i $(1 \leq i < k)$ を無関係なコネクションを含む大量のパケット系列の中から見つけ出す問題」といえる．

3.3 パケット系列

パケット系列は（世の中に多種出回っている）キャプチャソフトで各パケットの通過時刻とヘッダ(IP,TCP ヘッダ）を記録したものを使う．暗号化への対応やプライバシーの問題などから，我々はパケットの TCP データの内容はいっさい使わず，基本的には TCP データのサイズと通過時刻を使う．TCP データのコネクション開始からの累積バイト数は TCP ヘッダのシーケンス番号から分かる．シーケンス番号は 32-bit の正の整数で，コネクションの確立時にランダムな初期値が決まり，その後は送った TCP データ 1 バイト毎に 1 ずつ増える値が対応する．TCP ヘッダのシーケンス番号はそのパケットの TCP データの先頭のバイトのシーケンス番号である．

一般にチェインの上流のコネクションは下流のコネクションより先に始まり後に終わるので，上流のコネクションを探すならば，与えられたコネクションの時間を含む範囲でより多くの TCP データが流れたものに絞ることができる．

なお，一つのコネクションには互いに逆向きの 2 方向のパケット系列があるが，我々は今のところそれらをそれぞれ独立の系列として取り扱っている．

3.4 基本アイディア

コネクションの一方向のパケット系列について時刻とその時刻までのシーケンス番号の上限をそれぞれ X,Y 軸にとって平面上にプロットすると，図 2 のような単調増加のグラフになる．我々が想定しているのは，侵入者が踏み台ホストでスクリプトを走らせて自動的にコマンドを実行する状況ではなく，ある程度長い時間にわたって手でコマンドを入力してインタラクティブに作業する状況なので，各侵入行為毎にこのグラフは非常に特徴的な形になるはずである．よって異なるコネクションのグラフ同士で比べても，共にチェイン上にあった時間に対応する部分は似た形状になると予想できる．そこで与えられたパケット系列について他の系列との偏差を決める式を定義して，これが 0 に近ければその系列は同一チェイン上にあり，大きな値なら同一チェイン上にはないというようにする．

同じチェイン上の異なるコネクションの系列で

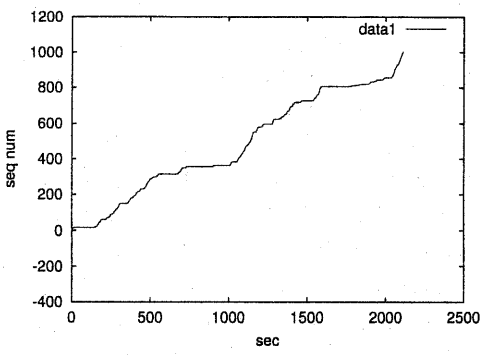

図 2: 時刻 - シーケンス番号のグラフ

はグラフの何が変わらず，また何が変わるのだろうか．まず，telnet や rlogin の通常の使い方をしている間は，チェイン上のどのコネクションでも流れるパケットの TCP データ部分はフロー制御やパケットの再送などを考慮すれば不変である．したがって，グラフの対応する部分の Y 軸方向の区間の長さは，シーケンス番号の増加分，つまり TCP データのバイト数であり，異なるコネクションのパケット系列間で等しい．なお，グラフの対応する部分の区間の開始位置は正確には分からないため，可能な全ての位置で比較する必要がある．

次に我々はシーケンス番号の上限を取るが，パケットの再送があったときは抜けていた TCP データが埋まるまでは先行して届いたデータもチェインの次のコネクションへは送られないことから，伝送の遅延時間はこの再送も含めた時間とみなされる．よって時刻が正確ならば，どの 2 つのコネクションでも上流に対する下流の相当する TCP データは，上流から下流の方向では後の，下流から上流への方向では先の時刻となり，矛盾は起きない．この遅延時間は分散が大きく，グラフは対応する部分の Y 軸方向の区間の位置をそろえればその区間は X 軸方向に歪んで伸びた形になると考えられる．

我々は侵入者がインタラクティブな操作をしていることを想定しているので，チェインを伝わるデータの平均の遅延時間はそれほどは大きくなく，普通は数百ミリ秒で大きくても 1,2 秒と想定する．平均で片道数秒以上も送れが生じたら侵入者にとっても操作能率が悪すぎるであろう．

3.5 パケット系列の偏差

与えられたパケット系列Aのグラフに対して，他のパケット系列Bのグラフを XY 平面内で上下左右に移動させ，Aと交わらない範囲で形が最もぴったり合うように近づけると，AがBと挟むX軸方向の隙間の平均は，BがAと同じチェイン上の系列だったならば小さな値になるであろうし，BがAと無関係ならば大きな隙間ができるため大きな値になるであろう．我々はこの“隙間の平均”をAに対するBの“偏差”として定義する．

例えば図3は，図2のdata1のグラフに対して，あるdata2というグラフをX軸方向の隙間が最も小さくなるように近づけたときの位置である．

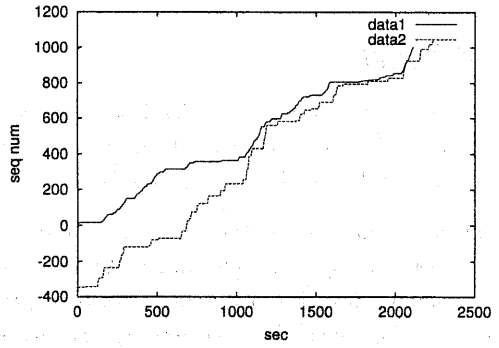

図 3: data1 に最も近づけた data2 のグラフ

いま，与えられたパケット系列Aはnパケットあり，第i番目のパケットのTCPデータのシーケンス番号を$a_{i-1}, a_{i-1}+1, \ldots, a_i-1$として（第$i$番目のパケットのTCPデータのサイズは$a_i-a_{i-1}$バイトである），シーケンス番号$s$ $(a_0 \le s < a_n)$に対応するTCPデータが含まれるパケットの通過時刻を$T(s)$とする．同様に，Aと比較するパケット系列Bはmパケットあり，第i番目のパケットのTCPデータのシーケンス番号を$b_{i-1}, b_{i-1}+1, \ldots, b_i-1$として，シーケンス番号$r$ $(b_0 \le r < b_m)$に対応するTCPデータが含まれるパケットの通過時刻を$U(r)$とする．このとき，Aに対するBの偏差を，$t(s,k) = U(s-a_0+b_k) - T(s)$，$\Delta = a_n - a_0$とおいて，次のように定義する．

$$\min_{0\le k<m}\left\{\left|\frac{1}{\Delta}\sum_{s=a_0}^{a_n-1}\left(t(s,k)-\min_{a_0\le s<a_n}\{t(s,k)\}\right)\right|, \left|\frac{1}{\Delta}\sum_{s=a_0}^{a_n-1}\left(t(s,k)-\max_{a_0\le s<a_n}\{t(s,k)\}\right)\right|\right\} \tag{1}$$

3.6 (平均－最小) 遅延時間

式(1)で与えた偏差は何を表しているのだろうか．まずα, βをそれぞれA,Bの時刻と正確な時刻との差とし，$\tilde{T}(s) = T(s)+\alpha, \tilde{U}(r) = U(r)+\beta$とする．AとBが同一チェイン上のコネクションのパケット系列だとすると，あるkで，Bの各シーケンス番号$s-a_0+b_k$ $(s = a_0, a_0+1, \ldots, a_n-1)$の表す1バイトのデータがそれぞれAのシーケンス番号sの表す1バイトのデータと同じ内容となる．$\tilde{t}(s,k) = \tilde{U}(s-a_0+b_k) - \tilde{T}(s)$とおき，簡単のため$\tilde{t}(s,k) \ge 0$の場合だけを説明する．$\tilde{t}(s,k)$はAで$s$の表す1バイトのデータの内容がチェインを伝わってBに到達するまでの時間である．ここで，式(1)のmin{}の中の上式は

$$\frac{1}{\Delta}\sum_{s=a_0}^{a_n-1}\left(t(s,k)-\min_{a_0\le s<a_n}\{t(s,k)\}\right) \tag{2}$$

$$= \frac{1}{\Delta}\sum_{s=a_0}^{a_n-1}\left(\tilde{t}(s,k)-\min_{a_0\le s<a_n}\{\tilde{t}(s,k)\}\right)$$

$$= \mu - \min_{a_0\le s<a_n}\{\tilde{t}(s,k)\} \quad \left(\mu = \frac{1}{\Delta}\sum_{s=a_0}^{a_n-1}\tilde{t}(s,k)\right) \tag{3}$$

である．式(2)は観測された時刻を使って計算された値なのに対して，式(3)は正確な時刻を用いて計った(平均遅延時間) － (最小遅延時間)である．よって，式(1)で与えた偏差はこれよりも小さい値である．

チェインの最初から最後まで流れるデータの平均遅延時間が高々1,2秒と想定すると，もしA,Bが同一チェイン上のコネクションのパケット系列ならば式(1)で計算される偏差も高々1,2秒である．

4 実験

式 (1) で計算されるパケット系列の偏差は，全く関係の無い系列に対しても小さくなる場合があるので，どの程度小さな値がどのくらいの頻度で現れるかを実験的に調べた．

我々は式 (1) で示したパケット系列の偏差を計算するプログラムを C 言語を用いて Linux(Red Hat 6.1) 上で実装した．パケットデータの読み書きは tcpdump[1] で記録されたものを扱えるように libpcap[1] を使用している．

我々はインターネットのあるバックボーン・ネットワークで 1 時間 tcpdump を使って記録されたパケットのヘッダーデータを用いて，そこから最低 1 分以上続き 60 バイト以上の TCP データが流れた telnet 又は rlogin のコネクションのパケットだけを取り出した．各パケット系列に対して他の全てのパケット系列との偏差をそれぞれ（合計で 18733 ペア）計算した．この値の $[0, 22)$ の範囲の分布が図 4 である．我々はこれ以外にも NLANR network traffic traces (`http://moat.nlanr.net`) から入手可能なデータ[2] でも実験してみたが，ほぼ同様の分布を得た．

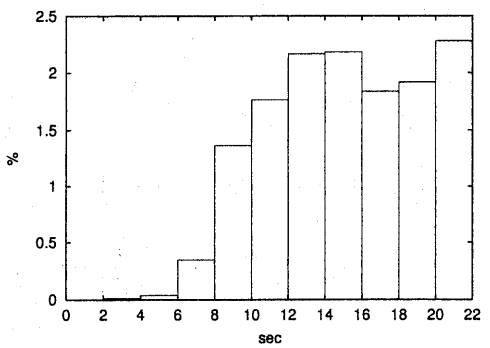

図 4: 無関係なパケット系列の偏差の分布

グラフを見ると，偏差が 8 秒より小さいものは全体の 0.5%以下しかなく，6 秒より小さくなることはほとんど無いことが分かる．したがって，この場合，大元から途中の踏み台までの片道の (平均遅延時間) − (最小遅延時間) が高々6 秒までのチェインならば，その踏み台の上流のコネクションのパケット系列を見つけることが可能である．もしもっと長い時間踏み台でパケット系列のデータが得られたならばこの上限は大きくなる．

[1] `ftp://ftp.ee.lbl.gov/` から入手可能

[2] データは the National Science Foundation NLANR/MOAT Cooperative Agreement (No. ANI-9807479), and the National Laboratory for Applied Network Research により提供

5 おわりに

本論文は，侵入者が長時間使った telnet や rlogin などのコネクションのパケット系列がある踏み台で得られたとき，そこより上流の同一チェイン上のコネクションを見つける手法を与えた．今後の課題は，ssh などで各チェインが暗号化されてデータサイズが変わっている場合にも対応することである．

参考文献

[1] H. T. Jung et al. Caller Identification System in the Internet Environment. In *Proceedings of the 4th Usenix Security Symposium*, 1993.

[2] S. Snapp et al. DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and An Early Prototype. In *Proceedings of the 14th National Computer Security Conference*, 1991.

[3] S. Staniford-Chen and L. T. Heberlein. Holding Intruders Accountable on the Internet. In *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, 1995.

[4] C. Stoll. *The Cukoo's Egg.* Doubleday, 1987.

[5] H. Tsutsui. Distributed Computer Networks for Tracking The Access Path of A User. *United States Patent 5220655*, Date of Patent Jun. 15, 1993.

[6] S. Wadell. Private Communications. 1994.

[7] 平. 俊明 and 宮. 聡. ネットワーク侵入経路追跡方式. **特開平** *10-164064*, 1998(平成 10) 年 6 月 19 日公開.