クロック遅れと進みを利用したホストフィンガープリンティングの実験

富永 祐樹† 田中 雄† 菊池 浩明†
東海大学情報理工学部情報メディア学科†

1 はじめに

近年，虚弱性を探索するポートスキャンが定常的に行われ，セキュリティの強化が必要不可欠なものとなっている．しかしながら，多くの攻撃では IP アドレスを偽装して不正パケットを送信するため，攻撃元の特定さえ困難になってきている．そこで，Kohno ら [1] はパケットから観測されるタイムスタンプの遅延から通信相手を識別する手法を提案している．この技術により，不正者の攻撃傾向を捉えることが出来れば，セキュリティ対策に活用することが期待できる．本稿では，[1] の手法を実験的に検証し，遅延の変動や OS による違いを明らかにする．実験データに基づき，本手法の精度として，識別可能なホストの台数を算出する．

2 遠隔デバイスフィンガープリント [1]

測定用ホストから被測定ホストに一定間隔 d で ICMP タイムスタンプを要求する．返信パケットを観測し，タイムスタンプ $T_0, \ldots, T_m$ を得る．観測ホストは適切な NTP サーバにより時刻を同期している．m 回の測定の真の時刻を $t_0, \ldots, t_m$ とする時，オフセット $y_i = T_i - T_0$，経過時間 $x_i = t_i - t_0$ と定める．ホストの skew a は，

$$y_i = ax_i \tag{1}$$

を満たす定数である．ネットワークの外乱などによる遅延が生じることがあるので，文献 [1] では，線形計画法に基づいて (1) 式を満たす上界値を定めている．a は時間に依らずほぼ一定であり，同一仕様のホストでもクロックの製造上小さな差が必ず生じる．この a を用いて，通信先が同一か否かを判定することができる．

3 評価実験

3.1 実験環境

自動的に ICMP Timestamp を観測するプログラムを開発した．Java と VisualBasicScript を使用している．本プログラムは，指定した時間 d で，定期的にタイムスタンプをログデータファイルに保存する．

3.2 実験目的

本研究では，(1) ホスト間の skew 差，(2)OS の違いによる skew 差，(3) 単一ホストでの skew の変動 を観測し，ホストの識別精度を明らかにすることを目的とする．

3.3 測定実験 1 ホスト間の skew 差

2007 年 8 月 29 日に，本学研究室の異なる仕様の 5 台のホストを対象に観測したクロックの遅延を図 1 に示す．

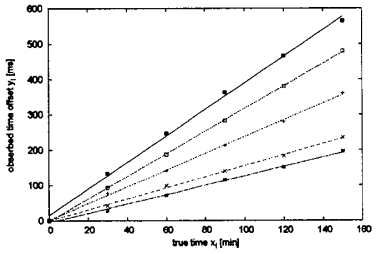

図 1: 観測時間の遅延 (オフセット)

図 1 から 5 台のホストあたりの遅れはほぼ一定であることがわかる．ホストの skew は，図 1 におけるグラフの傾き a で表されている．ホスト A の skew は，何回か測定を繰り返してもほぼ決まった遅れをする．

次に，2007 年 8 月 29 日に本学コンピュータ室の 91 台の同一機種を対象として，測定実験を行った．観測間隔は 30 分，観測期間は 150 分間である．結果を図 2 に示す．

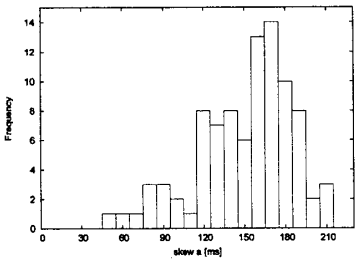

図 2: 91 台のホストの skew のヒストグラム

n 台の skew の平均値 $\mu(a_N)$ は 145[ms]，標準偏差 $\sigma(a_N)$ は 35 であった．従って，skew は 95%の確率で 75〜215[ms] に分布している．

Experiment on host fingerprinting based on clock skew

† Yuki Tominaga, Yu Tanaka, Hiroaki Kikuchi ({tomyu,hurry-go-round,kikn}@cs.dm.u-tokai.ac.jp), School of Information Science and Technology, Tokai University

3.4 OSの違いによるskew差

仮想OSを用いて，同一ホストでのOSの違いによるskewの差を調査した．観測期間は24時間である．skewの結果を表1に示す．

表1: OSとskew

OS	skew	R^2 値（誤差）
Windows2000	55.886	0.9967
WindowsXP	54.732	0.9942
Linux(Ubuntu)	653.32	0.9950

表1より，Windows2000とXPの差はほとんどなく，1[ms]程度の違いしか見られない．一方，WindowsXPとLinuxには大きな差があることがわかった．

3.5 測定実験3 単一ホストでのskewの変動

単一のホストA(Linux, Intel(R)Celeron, 1.8GHz, 472MB)について，30分間の測定を148回繰り返し，実測オフセット y'_i と理論値 ax_i との差を図3に示す．このホストAの30分間あたりのskewの平均値 $\mu(a_A)$

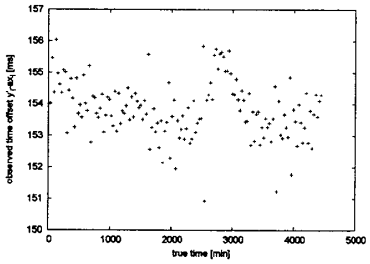

図3: skewによる個体差

は154[ms]，標準偏差 $\sigma(a_A) = 0.89$ であった．これは，ホスト間の標準偏差に比べて極めて小さい．

3.6 考察

本方式により，最大何台までのホストの違いを識別できるか検討しよう．まず，n 台のskew $a_1, \ldots, a_n$ の観測期間 T における平均が $\mu_{N,T}$，標準偏差 $\sigma_{N,T}$ の正規分布に従うこととする．a_i を取る確率変数を A_i と書く．n 台が独立に分布していると仮定 $(i.i.d.)$ とすると，T における標準偏差は，$\sigma_1 = \cdots = \sigma_n$ と表せる．例えば，実験1と3の結果は，

$$\sigma_{N,T} = 35 >> \sigma_1 = 0.89$$

の関係にある．ここで，$n = 91$，$T = 150m$ である．

ある2つのskew A_i と A_j の差を ΔA_{ij} とすると，正規分布の加法性より，ΔA_{ij} は平均0，標準偏差

$$\sigma_{\Delta A_{ij}} = \sqrt{2\sigma_{N,T}}$$

の正規分布になる．従って，この差 ΔA_{ij} が個々の分散 σ_i の幅と識別できない程小さくなる確率 P は，

$$p = P[|\Delta A_{ij}| < \sigma_i] = 2f(\sigma_i/\sigma_{\Delta A_{ij}}) - 1.0$$

で与えられる．ここで，f は正規分布の累積確率分布関数である．実験2，4の結果から，$p = 0.0072$ となる．

2台のホストのクロックが一致する確率が p の時，n 台のホストが全て異なるクロックを持つ確率は，いわゆる誕生日パラドックス[2]によって与えられることがよく知られている．50%以上の確率で，クロックの一致が生じる台数の上限を n^* とすると，

$$n^* = 1.18/\sqrt{p}$$

である．ここで $T = 150$ の場合，$n^* = 13.93$ であった．

図1から明らかな様に，n 台の標準偏差 $\sigma_{N,T}$ は期間 T に線形で増加する．一方，単一クロックの標準偏差 σ_i は T を上げても一定の値に収束する．それゆえ，T を広げると識別可能な台数 n^* は T に単調に増加することが予測される．そこで，$\sigma_{N,T}$ の線形性の仮定の下，観測期間 T について算出した識別可能台数 n^*_T の変化を図4に示す．

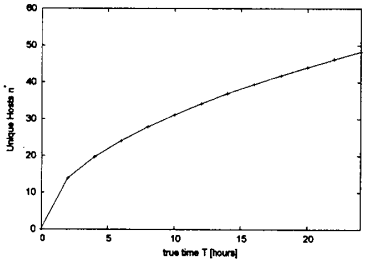

図4: 期間 T についての識別台数 n^*_T

図に示すように，n^* は T に対して増加するものの，その大きさには限界があることがわかる．

4 おわりに

[1]による，パケットから観測されるタイムスタンプの遅延から通信相手を識別する手法を再実験し，手法が有効であることを検証した．

今後の課題として，ICMPではなく，TCP Timestamp optionを利用する測定がある．

参考文献

[1] T. Kohno, A. Broidoand and KC. Claffy, "Remote physical device fingerprinting", 2005 IEEE Symposium on Security and Privacy, pp. 211-225, 2005.

[2] D. R. Stinson , "暗号理論の基礎",pp. 253 , 共立出版, 1996.