# Traceroute の発生間隔に基づく Crossfire 攻撃検知方式の誤検知率の解析

Analysis of False Detection Probability of Crossfire Attack Detection Based on Interval of Traceroute

仲原 愛美　　上山 憲昭
Manami Nakahara　　Noriaki Kamiyama
福岡大学 工学部 電子情報工学科
Faculty of Engineering, Fukuoka University

## 1. はじめに

DDoS (distributed denial of service) と呼ばれる，特定のサーバに大量のパケットを送り付けてサーバを機能不全とする攻撃が頻繁に発生している．DDoS 攻撃は攻撃ターゲットとなるサーバで到着パケットを観測することで検知・防御が可能である．しかし近年，特定のサーバではなく，複数のターゲットサーバ (TS) を含むターゲットエリア (TA) に至るネットワークのリンクを高負荷とすることで，TA へパケット到達不能とする Crossfire Attack (CFA) の問題が指摘されている [1][2]．CFA では攻撃対象がリンクであるためサーバにおける検知が困難であり，効率的・効果的な検知・防御方式の実現が課題である．CFA では攻撃対象リンクを選定するため攻撃に先立ち，攻撃に用いる大量のボットから TS に対し大量の traceroute が発生する特徴がある．そこで筆者らは traceroute の発生間隔から CFA の攻撃に用いられるボットを検知する方式を提案し，提案方式が正常ホストの誤検知を抑制しつつボットを高精度に検知することを示した [3]．しかし検知閾値の最適な設定法は未解決である．そこで本稿では，正常ホストの誤検知率の許容最大上限を設定し，それを満足する閾値を設定できるように，閾値に対する正常ホストの誤検知確率を簡易的な式で導出する．

## 2. Traceroute の発生間隔に基づく CFA 検知法

CFA では攻撃者はまず，多数のボットから TA 内の多数の公開サーバ (Web サーバ等) に対し traceroute を行うことで，TA 内のホスト宛てのフローの多くが経由する少数のリンク (攻撃対象リンク) を発見する．次に多数のボットから TA の周囲に存在する多数の公開サーバ (デコイサーバ) に対し traceroute を行い，攻撃対象リンクをフローが経由するボット・デコイサーバ組を選定する．そして選定したボット・デコイサーバ間に，検知されない程度の少量の正常なトラヒックを発生させることで攻撃対象リンクを過負荷にする．

[3] で提案した方式では，TS が短い時間内に複数のボットからの traceroute のターゲットとなる特徴を利用して，閾値時間 $T_d$ 内に連続して traceroute を受けたサーバを TS として検出してターゲットサーバリスト (TSL) に挿入し，TSL に含まれるサーバに対して閾値時間 $T_s$ 内に連続して traceroute を行ったホストをボットとして検知しブラックリスト (BL) に挿入する．そして BL に挿入されたホストが送信した traceroute の返信パケットを廃棄することでボットの攻撃リンク選定作業を妨害したり，またはボットが送信した攻撃トラヒックを遮断することで CFA を防御する．

任意のルータにおいて，発もしくは着 IP アドレスをキーとするハッシュテーブルを用いて BL と TSL を各々作成する．ルータが traceroute の ICMP パケットで TTL が 0 となるパケットを受信すると，発ホスト (着サーバ)IP アドレスをキーとしてハッシュテーブルに現在時刻を記録する．その際に，既に記録された値と現在時刻との差分が $T_s(T_d)$ 以下である場合に，その発ホスト (着サーバ) をボット (TS) として検出し，BL(TSL) に挿入する．ただし通常，1 回の traceroute において複数回の ICMP パケットが送信されることから，ハッシュテーブルには相手ホストの IP アドレスも記録し，相手が記録された IP アドレスと同じである場合にはボットや TS として検出しない．

## 3. 正常ホストの誤検知確率の導出

正常ホストが人気順位 $k$ のサーバ $S_k$ を traceroute の相手として選択する確率 $z_k$ は Zipf 分布に従うことを想定する．ボットの数と比較して正常ホストは圧倒的に多いことから，TSL への挿入確率の導出においてサーバにボットから到着する traceroute は無視する．全正常ホストからの traceroute 発生間隔の分布を $q(t)$ とする．測定期間内に j 回の traceroute を正常ホストから受けたサーバが TSL に挿入される確率 $w(j)$ は，$w(j) = 1-(1-q(T_d))^{j-1}$．全正常ホストからの traceroute 平均発生間隔が $\lambda_{n,all}$ のとき，正常ホストが測定期間 $M$ 中に j 回の traceroute をサーバ $S_k$ に対し行う確率 $g_k(j)$ は，平均が $B_k = Mz_k\lambda_{n,all}$ のポアソン分布に従うと仮定すると，$g_k(j) = {B_k}^j e^{-B_k}/j!$ となる．サーバ $S_k$ が TSL に挿入される確率 $W_k$ は，$W_k = \sum_{j=1}^{\infty} g_k(j)w(j)$ となる．

測定開始時において既に TSL が作成されていることを想定し，次に正常ホストの誤検知確率を導出する．$F_n(x)$ を正常ホストが $x$ 以下の時間間隔で traceroute を行う確率とする．1 組の連続する 2 回の traceroute によって，ある正常ホストが検知される確率 $D(T_s)$ は，$\boldsymbol{S}$ をサーバの集合とすると，$D(T_s) = \sum_{k1\in\boldsymbol{S}}\sum_{k2\in\boldsymbol{S},k2\neq k1} z_{k1}z_{k2}W_{k1}W_{k2}F_n(T_s)$ となる．$q$ 回の traceroute を行った正常ホストが検知される確率は $P_n(T_s) = 1-(1-D(T_s))^{q-1}$ となる．各正常ホストの traceroute 実施間隔の平均値を $1/\lambda$ とすると，ある正常ホストが測定期間中に $k$ 回の traceroute を行う確率 $h(k)$ は平均が $A = M\lambda$ のポアソン分布に従うので，$h(k) = A^k e^{-A}/k!$ となる．これらを用いて各正常ホストの誤検知確率は $p_n = \sum_{k=1}^{\infty} P_n(k)h(k)$ となる．

## 4. 性能評価

正常ホストの traceroute の発生時間間隔の分布 $F_n(x)$ として，WIDE のバックボーンネットワークで取得された公開パケットトレースデータを分析して作成した分布を用いた．また，$q(t)$ に関しても実際の分布を用いた．図 1(a) に $F_n(x)$ を示す．図 1(b) に 3 節で導出した式で計算された誤検知率と，計算機シミュレーションにより得られた誤検知率 FPR を各々，閾値 $T_s$ に対してプロットする．正常ホストの誤検知率は図 1(a) に示した $F_n(x)$ と似た傾向を示し，$T_s$ の増加に伴い FPR は増加するが，$T_s$ が 7 秒程度より大きくなると FPR の増加率が大きくなる．$T_s$ が 7 秒程度より小さい場合には解析による FPR の推定値はシミュレーション値とよく一致するが，$T_s$ がそれより大きな場合は誤差が見られる．誤差が生じた原因については，シミュレーションにおいて正常ホストの traceroute 発生間隔は実際の分布を用いているが，導出式では指数分布に従うと仮定したためであると予想される．そこで，ポアソン分布に従うと仮定した $g_k(j)$，$h(k)$ に関してシミュレーションによって得られた値を用いたグラフもプロットする．この場合，解析値とシミュレーション値の誤差はほぼ一致する．以上のことから，$T_s$ が 7 秒程度より小さい場合には，3 節で導出した誤検知確率式は良好な推定精度が得られることが確認できた．

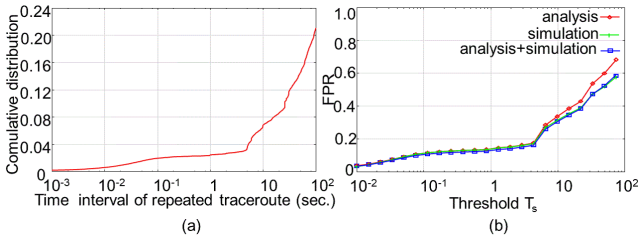


図 1: (a) 各ホストの Traceroute 実施間隔の累積分布，(b) 検知閾値 $T_s$ に対する正常ホスト誤検知確率

**謝辞**　本研究成果は，JSPS 科研費 18K11283 の助成を受けたものである．ここに記して謝意を表す．

**参考文献**

[1] W. Rafique, et al., CFADefense: A Security Solution to Detect and Mitigate Crossfire Attacks in Software-Defined IoT-Edge Infrastructure, HPCC 2019
[2] R. Rasool, et al., Cyberpulse: A Machine Learning Based Link Flooding Attack Mitigation System for Software Defined Networks, IEEE Access, 2019
[3] 仲原愛美, 上山憲昭, Traceroute の発生間隔に基づく Crossfire Attack の検知, 信学会 2020 年ソ大会, 2020 年 9 月