L-006

eBPF を用いたホストの通信履歴のアプリケーション単位の可視化
Application-level traffic visualization via eBPF-based packet capture

武田 風雅† 岡部 将也‡ 角田 裕§
Fuga Takeda Masaya Okabe Hiroshi Tsunoda

1. はじめに

イントラネットのセキュリティを確保する上で，各ホストの通信行動を詳細に把握することが重要である．特に，不審な通信の原因の特定を行うために，通信履歴をユーザ単位・アプリケーション単位で可視化することが望ましい．

しかし，従来のパケットキャプチャで直接取得できるのは，IP アドレスやポート番号などのネットワーク層およびトランスポート層（以下，L3/L4）の情報に限定される．そのため，キャプチャしたパケットがどのアプリケーションやユーザに紐づくものかを特定できず，前述のような粒度での通信履歴の可視化は困難であった．

この課題を解決するため，我々は eBPF (extended Berkeley Packet Filter) 技術を用い，パケットにアプリケーションのメタデータを紐づけて PCAPNG 形式で保存するシステムを提案した[1]．本研究では，このメタデータを活用し，ユーザ単位・アプリケーション単位の通信履歴をサンキー図[2]で可視化する手法を提案する．本手法により，従来の L3/L4 情報のみでは困難であった「誰が，どのアプリで，どこへ，どれくらいの量の通信をしたのか」を直感的に理解可能な形で可視化できる．

2. eBPF によるアプリケーションの識別

eBPF は，Linux カーネル内で発生する特定のイベントに対して，安全かつ柔軟に処理を追加できる仕組みである[3]．これにより，カーネルのソースコードを改変することなく，パケット処理の各段階で必要に応じて情報を安全に取得できる．

我々の提案システム[1]は，エンドホスト上で eBPF 技術を用いて送受信パケットをキャプチャし，パケットごとに対応するプロセスの ID（PID），プロセスの所有ユーザの ID（UID）を抽出する．さらに，PID を用いてアプリケーション名（App Name）を特定する．これら 3 つのデータをアプリケーションに関連するメタデータとしてパケットに付与し PCAPNG 形式で保存する．このメタデータは各パケットの Enhanced Packet Block が持つ opt_comment フィールドに格納されるため，Wireshark などの既存ツールでも参照可能である．提案システムが保存した PCAPNG ファイルを Wireshark で開き，パケットのメタデータを表示した様子を図 1 に示す．

このように，キャプチャしたパケットにアプリケーションに関連するメタデータを紐づけて記録することで，パケットの送受信に関係するアプリケーションやユーザの特定が可能となり，可視化に活用できるようになる．

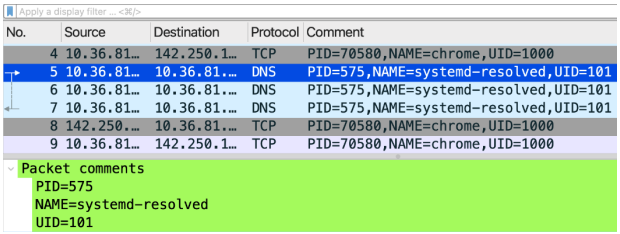

図 1　パケットのメタデータを表示した様子

3. 通信履歴のアプリケーション単位の可視化

サンキー図は，事象や要素をノードとして表現し，ノード間の関係や流量を帯の太さで可視化する手法である[2]．図 2 に，従来手法と提案手法の可視化のそれぞれでサンキー図のノードとなる要素を示す．

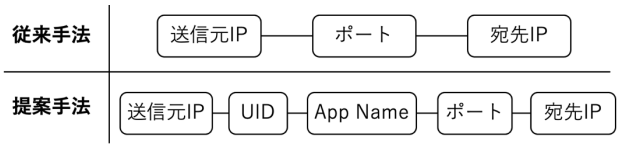

図 2　サンキー図の構成

従来手法では，パケットの送受信 IP アドレスとポート番号をノードとすることで，IP アドレスに対応するホスト間やポート番号に対応するサービス間のデータ流量を把握できる．しかし，例えば HTTPS サービスが使用する 443 番ポートのように，複数のアプリケーションが同一のポートにアクセスするようなケースでは，アプリケーションごとの通信量は把握できない．

一方，提案手法では，メタデータに含まれる UID と App Name をノードとして加え，UID → App Name → 宛先 IP という関係を表現する．この構造により，例えば HTTPS への通信であっても，それが Web ブラウザからの通信なのか，別のソフトウェアによる通信なのかを切り分け，それぞれの通信量を可視化できる．これにより，マルチユーザ環境において，どのユーザが操作するどのアプリケーションが，特定の宛先に対してどれだけの通信を行ったかを一目で把握することができる．

4. 提案手法の評価と考察

本節では提案するアプリケーション単位の可視化手法を実装し，従来のパケット情報ベースの可視化と比較することで，その有用性を確認する．

4.1 提案手法の実装

本手法での通信履歴の可視化手順を説明する．PCAPNG ファイルからパケットとそのメタデータを読み込み，前節で述べた提案手法のノード間の組み合わせごとにパケット

† 東北工業大学大学院工学研究科　Graduate School of Engineering, Tohoku Institute of Technology
‡ 東北工業大学情報サービスセンター　Information Service Center, Tohoku Institute of Technology
§ 東北工業大学工学部情報通信工学課程　Department of Information and Communication Engineering, Tohoku Institute of Technology

数をカウントする．そして，ノードのペアを帯でつないだサンキー図を生成する．ノード間の帯の太さはパケット数に比例させる．

なお，パケットキャプチャは観測対象のエンドホスト上で実施するため，当該ホストが送受信するすべてのパケットが観測される．本研究では，送信と受信を区別せずにカウントし，当該ホストを送信元とみなしてサンキー図を生成する．

可視化の処理の実装には Python を用いた．PCAPNG からの情報抽出には pyshark ライブラリを，サンキー図の描画には plotly ライブラリを使用した．

以上の実装により，メタデータが格納された任意の PCAPNG ファイルに対してアプリケーション単位で通信履歴を可視化できる仕組みを構築した．

4.2 可視化の結果の比較

図 3 に従来情報ベースの可視化結果を，図 4 に，提案形式に基づく可視化結果を示す．

使用データは， Linux ホスト上のユーザによる通信を我々の提案システムで観測し，PCAPNG ファイルとして記録したものから抽出した 160 パケットである．

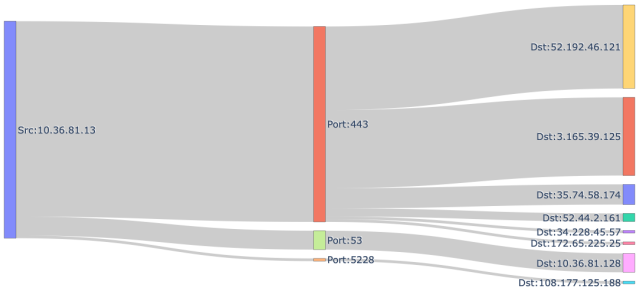

図 3　従来手法での可視化結果

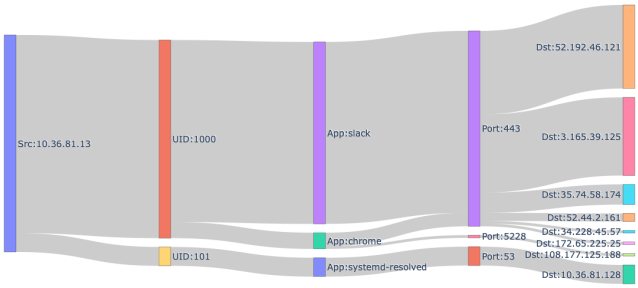

図 4　提案手法での可視化結果

図 3 より，10.36.81.13 は複数のホストの 443，53，5228 番ポートと通信しており，その多くは 443 番を占めていることがわかる．53 番や 5228 番はそれぞれウェルノウンポート，登録済みポートであり，またその先の宛先 IP アドレスがひとつであることから，通信行動がおおよそ推測可能である．例えば 53 番ポートとの通信であれば DNS による名前解決であり通信相手は DNS サーバであると推定できる．一方で，443 番ポートとの通信は，HTTPS 通信であることは考えられるものの，IP アドレスやポート番号といった情報だけでは，どのアプリケーションからの通信であるかを特定することは困難である．

図 4 の提案手法では，10.36.81.13 からの通信が UID1000 と 101 の 2 人のユーザによるもので，そのほとんどは UID1000 による通信であることがわかる．さらに，UID1000 のユーザの通信は， Slack と Chrome に分岐し，このユーザが 2 つのアプリを使用した通信を行ったことがわかる．次のポート番号のノードでは，Slack や Chrome に分かれたその多くが 443 番ポートを使用している．その後，宛先 IP アドレスに収束している．

このように，従来手法から階層の追加をしたことにより，通信行動の詳細な把握が可能になった．例えば，53 番ポートを使用した通信ではユーザ ID101 が割り当てられているユーザが systemd-resolve を使用して通信していたことがわかり，図 3 で読み取りが可能な情報に加え，通信を発生させたユーザ ID，および使用したプロセス名の把握が可能となった．443 番ポートを使用した通信については， UID1000 のユーザが Slack と Chrome で通信を発生させていることがわかり，図 3 で示した IP アドレスとポート番号のみを用いた従来手法の限界を改善することができた．

以上から， 通信行動の可視化において，UID や App Name といったメタデータを加えることで，ユーザ単位・アプリケーション単位での通信履歴の可視化が可能となった．

5. まとめと今後の課題

本研究では，我々が以前の取り組みで提案した，eBPF を用いたパケットにメタデータを付加するシステムから得られる情報を活用し，ユーザ単位・アプリケーション単位の通信の可視化手法の提案と評価を行った．

従来形式の可視化と比較し，送信元の識別粒度がホスト単位からユーザ単位へと向上し，さらにアプリケーション名単位での通信履歴の把握が可能となることを示した．

今後の課題としては， UID や IP アドレスが人間にとってややわかりにくい情報であることが挙げられる．そのため，をユーザ名やドメイン名などのわかりやすい文字列情報に変換することで，より通信内容の直感的な理解が期待される．また，提案手法の可視化におけるポート層は 443 番ポートのように複数のアプリケーションがアクセスに用いるため，宛先 IP アドレスが，どのアプリケーションから発生したものか識別困難になる．そのため，例えばポート層の位置を App Name 層より左に描画することや，宛先 IP アドレスの右側などに表示を工夫し，ノード間の最適化を図る．

謝辞

本研究の一部は JSPS 科研費 JP25K15120 の助成を受けたものです．

参考文献

[1] Masaya Okabe, Hiroshi Tsunoda, “An eBPF-based packet capture system with embedded application metadata for network forensics” , International Journal of Networking and Computing，to appear.

[2] 矢崎裕一，“サンキー・ダイアグラム（Sankey Diagram）”，visualizing.jp，2020-8-2，https://visualizing.jp/sankey-diagram/（参照 2025-06-08）

[3] eBPF コミュニティ，“eBPF とは？eBPF への入門と探究”，ebpf.io, https://ebpf.io/ja/what-is-ebpf/（参照 2025-06-09）．