[Work in Progress] 研究報告

通信のふるまいに着目した未知のアプリケーションによる通信の検知

市之瀬 樹生[1]　新城 靖[2]　三宮 秀次[2]　佐藤 聡[2]

Detection of communication by unknown application focusing on network traffic behavior

ネットワークの管理においてサイバー攻撃から組織を守ることは最優先事項の一つである。しかし、マルウェアに感染することを未然に完璧に防ぐことは困難である。

そこで近年はマルウェアの感染を前提とし、迅速に感染を発見し、対応する技術としてふるまい検知が注目を浴びている。ふるまい検知とはシステムログや通信、プロセス情報などからその端末のふるまいを分析してマルウェアに感染している兆候がないかを判断するという手法である。

しかしふるまい検知にはふるまいを監視するためのソフトウェアのインストールが必須であるという問題点が存在する。そこでこの問題を解決するための手法として通信情報のみから感染検知を行うことを提案する。

マルウェアの大半は感染行動や C2 サーバとの通信をはじめとした何らかの通信を行う。そのため、マルウェアの通信の意図を読み取ることができれば通信情報のみからマルウェアの感染を検知することは可能である。通信の意図を読み取るには DPI（Deep Packet Inspection）と呼ばれる通信情報を詳細に検査する方法を用いることが望ましい。しかし近年、セキュリティ意識の高まりにより HTTPS をはじめとした暗号化通信技術の普及が進んでいる[*1]。このため、DPI を行うことのできない通信が増えてきている。

そこで本研究ではパケットのペイロード部以外のヘッダ部や通信時間などのメタ情報からマルウェアの影響等による不審な通信を検出することを目的とする。この目的を実現するために機械学習により、ある通信がどのアプリケーションによる通信であるかを識別する識別器を作成する。より具体的に述べると TensorFlow の BasicLSTMCell を用いた RNN (Recurrent Neural Network) を作成し、入力に任意のポートで行われた一連の通信、出力にどのアプリケーションによる通信であるかを設定し、深層学習を行う。そして実際の通信においてアプリケーション識別器によって識別できない通信、つまり未知のアプリケーションによる通信があった場合に警告を行う。

しかし、機械学習を用いるには多量の訓練データが必要である。本研究で必要な訓練データは一般的な通信情報であるパケットデータに加え、その通信がどのアプリケーションによって行われた通信であるかというラベル付けが行われている必要があるがそのようなデータが公開されている例はなかった。また上記のような形式のデータを出力できる既存のパケットキャプチャソフトも存在しなかった。そのため本研究ではまず通信情報を収集することのできる Windows 用のパケットキャプチャソフトの開発を行っている。

パケットキャプチャを実現するために WFP (Windows Filtering Platform) と呼ばれる Windows ドライバーの API を用いる。WFP は従来の NDIS　(Netwrok Driver Interface Specification) フィルタなどのパケットフィルタリングに代わってより簡単な開発プラットフォームを提供している。この WFP にはレイヤが複数存在しており、どのレイヤを選択するかによってどのデータをフィルタリングするかを決定できる。例えば通信を行った IP や TCP/UDP のヘッダ情報については FWPM_LAYER_{INBOUND|OUTBOUND}_IPPACKET_V4 のレイヤを指定することで取得できる。また TCP 接続が確立されたときや非 TCP 通信が許可された際にフィルタするレイヤである FWPM_LAYER_ALE_FLOW_ESTABLISHED_V4 によってポート番号とプロセス ID の組み合わせを取得できる。このプロセス ID からプロセスパスを取得することでそのポートがどのアプリケーションによって使用されているのかを決定し、対応表を作成する。この対応表とクライアント側のポート番号からその通信がどのアプリケーションによるものかを決定することができる。

このパケットキャプチャソフトにより通常の利用によるプロセス情報とパケットの組み合わせを取得し、機械学習を実施する予定である。

[1] 筑波大学大学院博士前期課程システム情報工学研究科 コンピュータサイエンス専攻
[2] 筑波大学 University of Tsukuba

[*1] https://www.blog.google/topics/safety-security/say-yes-https-chrome-secures-web-one-site-time/