LSTMを用いたHTTPログ解析による マルウェア感染PC検知手法

一宮 秀星[1]　鳩野 逸生[2,a)]

概要：本論文では，保管されている HTTP 通信ログにおける各 PC の HTTP 通信における GET メソッドと POST メソッドの比を時系列データとして LSTM(Long Short-Term Memory) に学習させ，予測と実測を比較することにより通信傾向の変化を検知し，マルウェア感染の検知を試みている．本論文で提案する手法を，3 ヶ月間の HTTP 通信ログに適用した結果, 管理者により検知されていなかったものを含む 3 件のマルウェア感染の検知に成功した．

キーワード：マルウェア, 機械学習, 深層学習, HTTP 通信ログ, ログ解析

Detecting PCs Infected with Malware by HTTP Log Analysis Using LSTM

Shusei Ichinomiya[1]　Itsuo Hatono[2,a)]

Abstract: This paper deals with detecting PCs infected with malwares by HTTP Log analysis by using LSTM(Long Short Term-Memory). In this paper, time series data of ratio of number of GET methods and POST methos in each PC are learned by LSTM. By comparing the predicted and actual value, we try to detect PCs infected with malwares. Furthremore, we could detect 3 PCs infected with malwares, including an undetected PC by apply the method proposed in this paper.

Keywords: Machine learning, deep learning, HTTP log, log analysis

1. はじめに

情報を窃取したり破壊することを目的としたマルウェアが出現している. 特定の企業を狙ったマルウェアも存在しており, 標的型攻撃で利用される個別にカスタマイズされたマルウェアはシグネチャ型を用いるアンチウイルスソフトでは検知できないことが多い. またマルウェアの巧妙化により, 全てのマルウェアを検知することは不可能であるのが現実である.

この状況下において, ネットワーク管理者は, 各個人に呼びかけてクライアント側でのアンチウイルスソフトの更新やセキュリティーの規律を保つなどの予防措置によるマルウェアの感染を完全に防ぐための対策を行うだけでなく, 組織から外部への通信におけるエンドポイントでの防御や最低限に情報流出を抑えることが求められているが，十分とは言えない.

マルウェアが，外部へ情報を送信する場合には，HTTP 通信が使われる場合が多い[*1]．多くの組織で，組織内の計算資源を防御する目的で，必要最小限のポートのみに利用可能なポートを制限している場合でも，HTTP 通信は利用可能になるように設定されている場合が多いためであると推測している．一方で，HTTP 通信ログは情報セキュリティ情報インシデント対応などの目的で一定期間保存され

[1] 神戸大学大学院 システム情報学研究科
[2] 神戸大学 情報基盤センター
Information Science and Technology Center, 1-1 Rokko-dai, Nada, Kobe 657-8501 Japan
[a)] hatono@kobe-u.ac.jp

[*1] 論文執筆時 (2018 年)

ている場合が多い．神戸大学においても, インシデント発生時の調査や不正利用の監査を主な目的として, 学内から学外への HTTP 通信ログの情報を取得し保存している．

本論文では，神戸大学において保存されている HTTP ログに対して深層学習の一種である LSTM を適用することにより，マルウェアによる不正な通信を行っている PC を検知することを試みる．HTTP 通信ログを分析することによって，不正な通信を検出する試みとしては，帯刀らによる手法が提案されている [1] が，ヒューリスティックなアルゴリズムが用いられているため，汎用性が保証できない．また，機械学習によるマルウェア検知の研究も進められているが [2], [3], 進化するマルウェアによる攻撃の特徴を機械学習に継続して学習させる必要性があるため, 機械学習ベースによるアンチウイルスソフトはシグネチャ型のアンチウイルスソフトと同様に学習の為の大量の学習データ収集が可能な場合にしか適用できない．

本論文では, 神戸大学における HTTP 通信ログを用いることにより，GET メソッド数と POST メソッド数の比を一時間毎の時系列データとして学習し，過去 24 時間の通信記録を学習することによって得られた予測値と実測値を比較することにより, マルウェアが情報流出させている状態を検知することを試みる．マルウェアによる情報送信が数多く行われているような状況では，GET メソッド数と POST メソッド数の比が通常に比べて変化すると考えられるためである．本手法においては，学習において不正であることが判明しているデータを必要としないため，適用範囲が広くなることが期待される．

2. HTTP 通信ログの取得

2.1 HTTP 通信ログ取得位置

本論文においてログ通信収集した当時のネットワーク概略を図 1 に示す. 学内からの通信がファイアウォールを通る直前に設置しているロードバランサのポートをミラーし, それらのポートを解析サーバから通信モニタリングソフトウェア tshark を用いることにより HTTP 通信ログを取得している. 基幹ルータからロードバランサは 10Gbps で接続されているが, ミラーポートは 1Gbps であることと, 解析サーバの性能を考慮するとかなりのパケットを取りこぼしていることが予想される．

2.2 HTTP 通信ログ取得内容

神戸大学で保存されている HTTP 通信ログには，時刻, ソース IP, request method, 通信先 URL, 相手先 IP, リファラー, User-Agent が含まれており，タブ区切りのデータとして保存されている．通信ログの例を図 2 に示す [1].

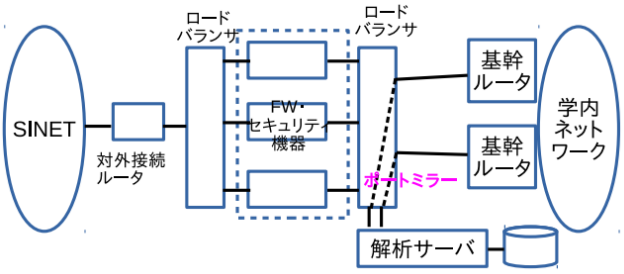

図 1　HTTP ログ取得機構

1514288319.783876000 0 1xx.y0.zz.aa,TCP_MISS/ GET,http://X.url.b.com:80/T/108/-Sc5u4W-6RbtnjONdNpZJW05fBo9knxeb1rCwmJZAP2LYFPNZk-F-y6kjcVYiAm10wkg2AzocQlJM6Fxuw9E6v-B_8OwQ2-zHRaDowHm4XU= - DIRECT/23.46.140.xx text/html TMMM

1514288320.672256000 0 1xx.y0.zz.bb TCP_MISS/ GET,http://X.b.com:80/T/88/-Sc5u4W-6RbtnjONdNpZJW4GLHcvVa1-C7G83cR8ir2IOfOefFfKj83VGIzAv7rUbq6lUl39KjMjF2V_2MjJ0Q== - DIRECT/23.46.140.xx text/html TMMM

図 2　HTTP ログの例

3. 不正な通信を行っている PC の検知手法

PC がマルウェアに感染した場合，PC のキーボードの入力情報，クリップボードの情報，内部の重要なファイルなどが外部のサーバに向けて送信されるという事象がよく観測される．これは，現在のマルウェアの多くが何らかの情報を PC から盗み出すことを目的としているためであると考えられる．

一般のユーザが，PC からブラウザ等を用いて HTTP 通信を行う場合, GET メソッドの通信が多く観測され, POST メソッドの通信は比較的少数である．PC がマルウェアに感染し，バックグラウンドジョブによる情報送信が多数発生しているような状況では，通常の利用時と比較して多くの POST メソッドの通信が行われると推測される．本論文では，このような状況の下で，マルウェア感染によって情報が外部に送信されているという状況を, 深層学習の一種である LSTM(Long Short-Term Memory) によって検知することを試みている．

3.1 LSTM の概要

LSTM は, 時系列データにおける短期だけでなく長期に渡る依存性を効率的に学習するために開発されたリカレントネットワーク (RNN) の一種であり [4][5], RNN の中間層のユニットを (1) 長期依存性を学習するための CEC(Constan Error Carousel), (2) 依存性がない入出力による重み衝突を解消するための入出力ゲート, (3) 忘却を実現するための忘却ゲートを導入した LSTM ゲートに置き換えたものである．本論文では，特定の IP アドレスを持つ PC からの

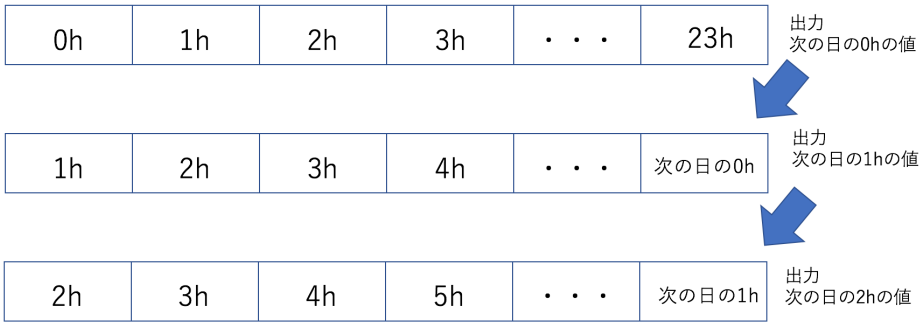

図 3 時系列データの学習

HTTP 通信における, HTTP GET と POST の傾向をログデータから学習するために LSTM を用いる.

3.2 LSTM による通信傾向の学習

本論文においては，情報流出が伴うようなマルウェアに感染し, HTTP POST メソッドにより外部へ PC 内のデータが送信される場合を考える. この場合，観測される HTTP GET メソッドに対して，HTTP POST の数は，相対的に大きく増加すると考えられる. このような仮定の下で, LSTM で通常時の傾向を学習し，LSTM による予測と実測値が大きく変化した場合に，マルウェアに感染している可能性がある PC であると判定する[*2]. ただし，本仮定は，ネットワーク内では, 1 つの IP アドレスにつき 1 台の端末が接続され, かつ固定されていることを前提としている. また，パケットの取りこぼしによるログの欠損が存在したとしても，欠損はランダムに発生すると考えられる. 従って，POST と GET 数の割合には大きな変化がないと考えられるため，本手法の適用には大きな問題とはならないと思われる. 対象となる URL のアクセス数が非常に少ない場合には，POST と GET 数の比にログの欠損が大きな影響を与える場合もあるが，本論文では，このような場合は対象としていない.

本論文における学習データは, 通信ログデータに出現する各 IP アドレス毎に HTTP Requst における HTTP POST と GET メソッド数の割合を 1 時間毎に求め，時系列データを生成し, LSTM における学習データとする. 現時刻を t とし，時刻 t における IP アドレス i の HTTP POST と GET メソッド数の割合を P_t^i とする. このような学習データを用い，過去 24 時間分の時系列データ $P_t^i, P_{t-1}^i, \ldots, P_{t-23}^i$ を入力とし，P_{t+1}^i を出力とする LSTM を構成し，時系列データの学習を行う.

以下に, LSTM による学習アルゴリズムを示す.

LSTM による学習アルゴリズム

Step 1: 学習を行う期間に観測した HTTP 通信ログにおいて出現するすべての IP アドレスについて HTTP POST と GET メソッド数の割合の時系列情報 P_*^* を求める. また, 観測された IP アドレスの集合を I とする.

Step 2: 時刻を学習データの観測開始時刻 $t = t_0$ とする.

Step 3: すべての IP $i \in I$ に対して, Step 4 以下の手順を行う.

Step 4: 時系列データ $P_t^i, P_{t+1}^i, \ldots, P_{t+23}^i$ を入力データ, P_{t+24}^i を教師データとして，逆伝播による重みの調整を行う.

Step 5: $t = t + 1$ として，t が学習データの観測期間内の間 Step 4 を実行する.

図 3 に，時系列データの学習の様子を図示する.

3.3 不正通信の検知

学習済みの LSTM を用いて，以下の手順でマルウェアに感染している可能性がある PC の検出を行う.

Step 1: すべての t および $i \in I$ について，LSTM による予測値と実測値のずれを計算し，一日毎のずれの和の平均を求める.

Step 2: 期間内の各日毎に，ずれの和を計算し，値が 1.5 倍以上上回っている IP アドレスを, マルウェアが感染している可能性がある PC とする.

Step 3: マルウェアに感染している可能性がある IP の HTTP ログの中の POST メソッドの通信先の URL を検査し，不正な通信先が含まれていないか調べ，不正な通信先が含まれていないかを「VirusTotal」サイトを用いて検査する.

Step 5: 不正と判定されなかった場合，実測値を用いて LSTM の学習を実施する.

本手法は, マルウェア検知において膨大なログの中からマルウェアに感染し情報流出の疑いがある PC の絞り込みを行っている. まず, ずれの平均が 1.5 倍以上であるものの絞り込みを行い, その後, 絞り込みが行われた IP と日時を

[*2] 情報流出を伴わず，遠隔操作のみが目的の場合は本仮定に合致しない

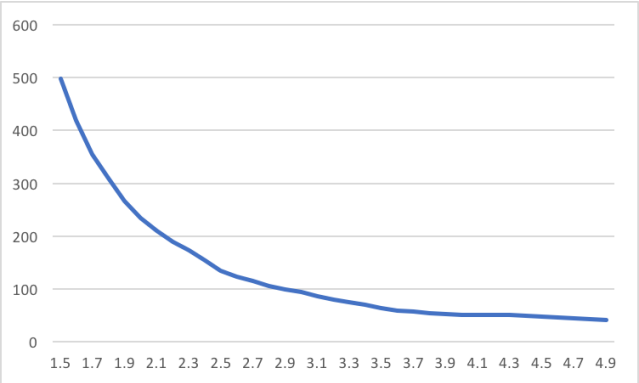

図 4 ずれの値を変化させることによる絞り込み数

元に POST のホスト先を外部サイトである「VirutsTotal」[6] の API を用いて与えることにより最終的なマルウェア感染の有無を調査している．本手法は，通信のパターンの変化を用いてマルウェアによる通信を絞り込む手法であるため，特定された URL が本当にマルウェアのものであるかを最終確定するには別途調査が必要である．今回ずれの平均を 1.5 倍位以上と設定したのは, 保管されているログの量から絞り込んだ数が現実的な時間内に VirusTotal の API で調査可能数になるように設定した為である．**図 4** に示すように，本手法を適用する場合, 設定値を上げることにより絞り込み数を少なくすることが可能であるが，一方で情報流出量が大規模とは言えない場合を見逃す可能性が高まる可能性がある．

4. 評価実験

4.1 LSTM における学習

本手法の適用に際して, 約 1000 台のコンピュータが接続されていると推定される部署の HTTP 通信ログ 2014 年 7 月 20 日–2016 年 7 月 20 日を対象に PC1 台ずつの学習データを作成し, その後の 3 ヶ月分のデータをテストデータとした．学習データ中で，対象となる通信先のホストは約 23 万件であった．LSTM のモデルの作成には Python のライブラリである「Keras」[5] を用いている．計算には，nVIDIA P600 ボードを有した PC における GPU 演算によって行っている．また，ひとつ IP アドレス (すなわち一台の PC) の学習に，約 30 分要した．ただし，隠れ層数は 300 であり，24 単位時間 (本論文の場合は 24 時間) 内の隠れ層に対して学習を行う．

4.2 学習状況の評価

対象となる PC の中の一台の学習の収束状況を**図 5** に示す．縦軸の Loss は，予測値と実測値との平均二乗誤差, 横軸の epoch は，学習反復回数を示す．

図 5 から, 学習回数を重ねる毎に平均二乗誤差が減少しているところから，少なくとも 300 回学習した時点まで 学習が正常に進行しているものを思われる．他の IP アドレスに対しても同様な傾向を示すことを確認している．

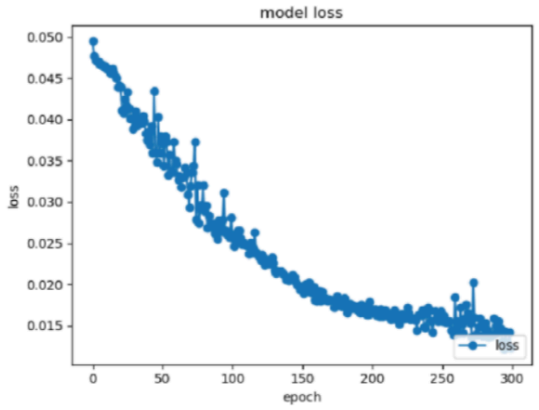

図 5 学習の収束状況の例

表 1 マルウェア感染 PC における予測値と実測値の一日分のずれの例

日付	1 日分のずれ
7/21	4.3
7/22	5.3
7/25	5.7
7/26	5.6
7/29	8.1
8/2	6.6
8/3	7.4
8/4	6.5
8/5	6.6
8/8	6.8
8/9	5.8

4.3 既知のマルウェア感染 PC を用いた評価

2016 年 7 月における情報流出が疑われるマルウェア感染 PC に対し，2014 年 7 月 20 日から, 2016 年 7 月 20 日までのログデータを用いて学習した LSTM による予測値と, マルウェア感染が確認された PC における実測値とのずれを**表 1** に示す．

2016 年 7 月 21 日から 2016 年 7 月 26 日の間に比べて, 2016 年 7 月 29 日から 2016 年 8 月 8 日の間はずれが 1.5 倍以上になっている事が確認できる．実験に用いたログを 2016 年 7 月 29 日から 2016 年 8 月 8 日の間に不審な通信先への通信が行われていないか検査を実施した．その結果発見されたウイルスに感染したと思われる 2016 年 7 月 29 日の通信先を**図 6** に示す．図 6 に示すように,「http://XXXX75.com/」への通信先が異常に多くなっていることが確認できる. ただし，http://XXXX75.com/は, 実際のホスト名の一部を変更して表示している. これらの通信先は, VirutsTotal でマルウェアと判定されることを確認している．

また, 予測値とのずれが平常に近い値に戻っている 2016 年 8 月 9 日の通信先を調査すると,「http://XXXX75.com/」

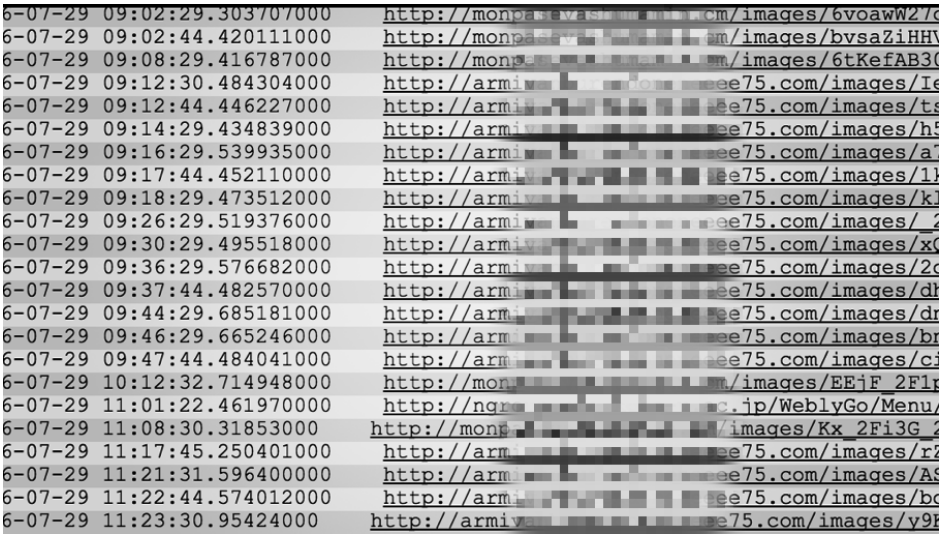

図 6 既知の情報流出の通信先の HTTP ログ

への通信が無くなっており, マルウェアの活動が行われず通常の通信が行われていたと推測される.

以上のように，2016 年 7 月に発生した既知のマルウェア感染 PC を本手法により検知できることを確認した．同種類のマルウェアに感染した場合，ほぼ同じ動作すると考えられるため，少なくとも 2016 年 7 月に神戸大学において感染が確認されたマルウェアは同様に検知できるものと思われる.

4.4 3 ヶ月間のログデータへの適用

4.1 節において学習後の LSTM を，学習期間後の 2016 年 7 月 21 日–2016 年 10 月 30 日のログデータに適用した結果，500 件程の不正な疑いがある通信が検出された．さらに，詳細に通信ログを確認したところ，2 件のマルウェア感染が疑われる PC が存在したと判定することができた.

1 件目の IP アドレスにおける 2016 年 10 月 24 日の通信先を**図 7** に示す.

図 7 からランダムな文字列の URL に多数通信していることが確認できる. 外部の判定サイトである VirutsTotal を用いて通信先を調べたところマルウェア判定となった. この URL の末尾の拡張子は「.bmp」であり, 2016 年 7 月 29 日に発生したマルウェアによる不正通信の際にも, ホスト名が異なるが末尾が「.bmp」であったため類似のマルウェアに感染したと推測される. 本件については学内ネットワーク管理者は把握済みであったものの, 4 日間ほど予測値が異常であった事から対処するまでに 4 日程度かかっていることが確認できている. 本手法を適用した際には不正な通信は始まった日に異常検知を発見したため, 1 日で情報流出を止めることができた可能性が高いと考えられる.

2 件目の IP アドレスにおける 2016 年 7 月 22 日における通信先を**図 8** に示す．通信先である「bis.XXXX.com」を外部サイトである malwares.com[7] にて検索した結果, 過去にマルウェアであることが判明した． ログからはどのような情報が送信されたかは判断することができないが, アクセスする頻度を鑑みると感染したパソコンから何らかの情報が送信されていることが推定される． 本件は学内ネットワーク管理者も把握できていなかった案件であり, 直ちに報告を行った. ログだけではどのような情報が流出したかは不明であるが, マルウェアによる不正通信であった可能性は非常に高い.

本手法を学内ログに対して 2016 年 7 月 21 日–2016 年 10 月 30 日を対象に適用した結果, 未知の情報流出の痕跡を 2 件発見することができた.HTTP POST と GET の割合の値を学習して異常値を検出するため, POST の数が増える大規模な情報流出に対しては有効であることが推察できる. また, モデルを一日ごとに学習しながら検知することが可能であるため，大規模な情報流出の兆候が起こった際には 1 日で発見できる.

一方で, 図 7 や図 8 のように, 不正な通信が何件も連続して現れた場合には検知することが可能であるが, 小規模な情報流出の場合に POST の数が少なかったりすると本手法では発見することは困難である.

5. 終わりに

本論文では，HTTP 通信ログを用いることにより，各 IP アドレスごとに HTTP GET および POST メソッドの発行回数の比を時系列データとして LSTM で学習し，LSTM による予測値と実測値の差が大きくなった場合を検出することにより，情報流出を伴うようなマルウェア感染を検出することを試みた.

神戸大学における 2016 年 7 月 21 日–2016 年 10 月 30 日まで HTTP ログデータに適用した結果，管理者が検知していない 1 件を含む 3 件のウイルス感染が疑われる PC を検出することができた．しかし，1 つの IP について 2 年間

図 7 本手法により発見された 1 件目の不正な通信先への HTTP ログ

図 8 本手法により発見された 2 件目の不正な通信先への HTTP ログ

のデータを用いて学習を行うと約 30 分の学習時間を要するため，全学の全 IP を対象とするためには学習の高速化を行う必要がある．また，大量の POST メソッドを伴わない場合は，本手法においては検出困難であることから，他の手法と組み合わせることにより検出精度の向上を測る必要がある．また, 近年は HTTP 通信の SSL 化が進んでおり, ログ取得が困難になっている. 今後 SSL 化が進行した場合，本手法の適用が困難になることが予想される．今後は, IDS の出力情報の併用などにより改良していく必要があると考えられる．

参考文献

[1] 帯刀, 鳩野, “HTTP 通信ログ解析を用いた不正プログラム感染 PC 検知の試み,” インターネットと運用技術シンポジウム 2015 論文集,2015,79-85
[2] 小池, 中谷, 萩原, 厚井, 高倉，吉田, “ベイズ学習アルゴリズムを用いた未知のコンピュータウイルス検知手法”, 情報処理学会論文誌, Vol. 46. No. 8 , 2005, 1984-1996
[3] Joshua Saxe, Konstantin Berlin, “A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys,” arXiv:1702.08568, Cornell University Library, 2017
[4] F.A.Gers, J. Schmidhuber, F. Cummins, Learning to Forget: Continual Prediction with LSTM, Neural Computation, Vol 12, 2000, 2451-2471
[5] 巣籠, 詳解 ディープラーニング –TensorFlow・Keras による時系列データ処理–, マイナビ出版, 2017
[6] virustotal https://www.virustotal.com/(2018)
[7] malwares.com https://www.malwares.com(2018)