通信パケットの記録からの Web を介する攻撃の再現 —メッセージフローの自動再現とログの再取得の検討—

奥田裕樹[†1]　福田洋治[†2]　井口信和[†2]

概要：著者らはこれまでインシデント対応における調査活動を支援するためのフォレンジック支援システムを開発してきた．著者らのシステムは通信パケットの記録から Web サイトを復元し再現端末からこれにアクセスし Web を介した攻撃を再現して起こった事柄を再観測できる．本稿では通信パケットの記録からセッションを抽出しそのメッセージフローを自動再現しながらこれと連動して再現端末で OS やアプリケーションのイベントログ，動作ログを再取得する機能について検討する．この機能は大量の通信パケットの記録から Web を介した攻撃に該当する可能性のあるところを抽出する過程を支援する．

キーワード：インシデント対応，Web を介した攻撃，パケット，Web サイトの復元，メッセージフローの再現

Reproduction of Attacks via Web by Using Captured Packets - On Message Flow Reproduction and Log Acquisition-

YUKI OKUDA[†1]　YOUJI FUKUTA[†2]
NOBUKAZU IGUCHI[†2]

Abstract: The authors have developed a system that supports investigation of attack communication data and malware activities in Web based attacks which is one of the dominant ways of sending malware to client PCs. This system reproduces HTTP requests, responses and behavior of malicious website related to the Web based attack from the raw packets of the communication at the time of the incident occurs. In this manuscript, we consider the function to extract the HTTP session from the raw packets and reproduce the message flow, and reacquire event log and operation log of the OS and application in the client PC. This function supports the process of extracting the raw packets corresponding to attacks via the Web from the large number of captured raw packets.

Keywords: Incident Response, Web based attack, Raw packets, Reproducing website, Message Flow Reproduction

1. はじめに

組織内の端末が Web を介しマルウェアに感染して，情報の漏洩や金銭の要求，不正な遠隔操作が行われるなどの，インシデントの報告が増加の一途を辿っている．インシデント対応では，根絶と復旧，再発予防の観点で，当時起こった事柄を把握することが求められる．しかしながら，端末で履歴が記録されていない，または消去・攪乱されると，その起こった事柄の全容を把握できない場合がある[1].

これまで著者らは，インシデント対応における調査活動を支援するためのフォレンジック支援システムを開発してきた[2]．著者らのシステムは，通信パケットの記録から Web サイトを復元し，再現端末からこれにアクセスし Web を介した攻撃を再現して，起こった事柄を再観測できる．通信パケットの記録とは，FW などの境界で取得した raw パケットを PCAP 形式で記録したものである．

本稿では，通信パケットの記録からセッションを抽出し，その HTTP メッセージフローの自動再現を行うと同時に，再現端末上で OS やアプリケーションのイベントログや動作ログを再取得する機能について検討する．この機能は，再現対象の HTTP メッセージフローを再現しながら，指定したロガーを起動する．また，自動で起動したロガーで収集したログの保存ファイルと再現時に発生した HTTP メッセージフローをグラフ化し利用者に提示する．これにより，利用者は再現対象となる HTTP メッセージフローを選択するだけで，当時行われた Web サイトへのアクセスとその際の Web クライアントの挙動を再現することができ，その際に再現端末上で起こった事柄を記録できる．収集した OS やアプリケーションのログと再現時の HTTP メッセージフローの情報を提示することにより，インシデント対応の調査の場面において通信パケットの記録から Web を介した攻撃に該当する部分の抽出を支援する．

2. Web を介した攻撃の振る舞いを再現するフォレンジック支援システム

著者らのシステムは，図 1 のように，疑似 Web サイトと誘導 DNS サーバから構成されており，復元した Web サイ

†1 近畿大学大学院
Graduate school of Kindai University
†2 近畿大学
Kindai University

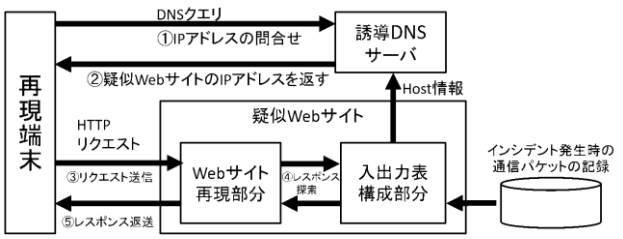

図 1　Web を介した攻撃の振る舞いを
再現するシステムの構成と動作

Figure 1　Components and flow of a system to reproduce the

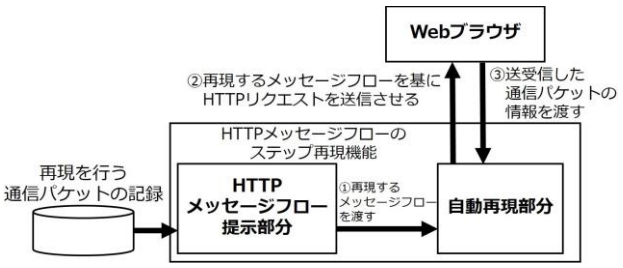

図 2　ステップ再現機能の構成と動作

Figure 2　Components and flow of step reproduction function

トに対し，仮想環境上の Web クライアントからアクセスすることでアクセス時の Web クライアントの挙動がそのまま再現される[2].

疑似 Web サイトは，入出力表構成部分で読み込んだ通信パケットの記録から HTTP セッションを抽出し, HTTP リクエストとそのレスポンスの表を作成, 管理する. その後，Web サイト再現部分で Web クライアントからのアクセスを受け付け，入出力表を基に HTTP リクエストに対するレスポンスを返す．誘導 DNS サーバは疑似 Web サイトの IP アドレスと復元する Web サイトのホスト名の対応を A レコードとして持つ．再現端末からのアクセスに伴う名前解決時にこの誘導 DNS サーバを参照させることで疑似 Web サイトにアクセスを誘導する．

Web を介した攻撃の振る舞いを再現するシステムの動作の流れを以下に示す.

① 通信パケットの記録からリクエストとそれに対応するレスポンスを抽出する
② リクエストとレスポンスの対応を入出力表として入出力表構成部分で保持する
③ リクエストのホスト名と疑似 Web サイトの IP アドレスを対応付けて A レコードとし誘導 DNS サーバに登録する
④ 再現端末からの HTTP リクエストを疑似 Web サイトの Web サイト再現部分で受信する
⑤ 入出力表を基に受信したリクエストに対応するレスポンス探索する
⑥ 対応するレスポンスがある場合，アクセス元の Web クライアントに対し返送する
対応するレスポンスが無い場合，アクセス元に対し 404 エラーを返送する

誘導 DNS サーバが仮想環境からの名前解決を行うことで通常のアクセスを疑似 Web サイトに誘導する. これにより, 通常の Web 利用と変わらずに当時アクセスされた Web ページとその描画に起因する PC の挙動が再現できる.

再現端末は仮想化技術を用いて被害を受けた可能性がある端末と同じソフトウェア環境を構築して使用することを想定している．これにより，実際にマルウェアが設置された場合においてもホスト OS への影響を防ぎ，様々な環境の構築や Web クライアントからの試行を容易にする.

再現端末には，再現対象となる通信パケットの記録に含まれる HTTP メッセージフローのステップ再現を行う機能がある．この機能は図 2 のように HTTP メッセージフロー提示部分と自動再現部分から構成されており，再現を行う通信パケットの記録に含まれる HTTP メッセージフローを, Web クライアントを用いて自動で再現する.

HTTP メッセージフロー提示部分は, 再現対象となる通信パケットの記録を読み込み，その中に含まれるセッションを抽出し，IP アドレスごとに分割したうえでその HTTP メッセージフローの情報を通信パケットの送信方向や通信パケットの概要を利用者に提示する．この提示した HTTP メッセージフローと通信パケットの概要を基に再現を行う HTTP メッセージフローを選択する．自動再現部分は，HTTP メッセージフロー提示部分で提示した HTTP メッセージフローの中から利用者が指定した HTTP メッセージフローを，Web クライアントを操作して再現させる．自動再現部分では，Web クライアントをコントロールするために Selenium WebDriver[3]を用いる．Selenium WebDriver を用いることにより, アプリケーション側から任意の Web クライアントを操作することが可能になる．また，自動再現部分は，再現を行う HTTP メッセージフロー内に含まれる通信パケットが正しく送受信されているかをチェックする．通信パケットのチェックの流れを以下に示す.

① Web クライアントを操作して HTTP リクエストを送信する
② Web クライアントに到着したレスポンスを再現対象の HTTP メッセージフローのものと比較する
③ レスポンスがすべて到着すると，次に送信されるべき HTTP リクエストが送信されるか確認する
④ 5 秒以内に次の HTTP リクエストが送信されない場合，Web クライアントを操作し HTTP リクエストを送信する（送信されると，手順②に戻り，再現が完了するまでこれを繰り返す）

チェックした結果は利用者に提示した HTTP メッセージフローの情報と共に表示する.

Web を介した攻撃を再現するシステムとメッセージフロ

ーのステップ再現機能を用いて Web クライアントから復元した Web サイトへ自動でアクセスさせ，当時発生した事柄を再現し，再現端末上で動作させている通信パケットやプロセスの挙動を観測・記録するためのツールのログなどから，当時どのようなことが起きていたのかについての調査を支援する．

このシステムで，Web ページの改竄による誘導やメールに記載された URL を用いて誘導するもの，サーバ側スクリプトの脆弱性を利用して誘導するもの，Web 上における詐欺行為を用いて誘導するものの 4 種類の誘導の手段からつながる Drive-by Download 攻撃を再現できることを実験により確認した[2]．Web ページの改竄による誘導では，JavaScript を用いる方法，iframe タグを用いる方法，レスポンスの Location ヘッダを用いる方法のいずれも再現できる．またサーバ側スクリプトの脆弱性を利用するものとして Stored XSS を用いた方法も再現できる．Web 上における詐欺行為による誘導では ClickJack 攻撃を用いる方法が再現できることを確認している．現在のシステムでは，クライアントサイドスクリプトにより，アクセス先が通信パケットの記録に含まれるものと異なるものになり当時と異なるリクエストが送信された場合は再現ができないが，これについては今後の課題としたい．

本システムを用いて攻撃を再現し，マルウェアが設置・実行されると，再現端末上でその挙動を観測することができる．このとき，観測可能な挙動について，岩崎らによる APT の分類と対応策についての一考察[4]で APT のプロセスにおける第 3 段階で実行される攻撃行為の分類に基づき観測可能であるか整理した．攻撃行為は 6 つの種類に分類されている．このうち，バックドアの設定，システム内部の調査と痕跡の削除については再現端末内部で完結するため再現・観測が可能であると考えられる．また，システム外部からの送信・受信に関しては，使用されるプロトコルが HTTP であるかつ再現端末もしくは設置されたマルウェアが Web クライアントとして動作する場合は，本システムが対応できると考えられる．

HTTP 以外のプロトコルを用いる場合は現状のシステムで再現することはできない．また，再現端末もしくは設置されたマルウェアがサーバとして動作する場合については，現状のシステムに能動的に通信パケットを送信する機能が無い為再現することができない．ネットワーク内部への拡散については，本システムでは，再現端末は 1 台での利用を想定しており，LAN に相当する再現端末側のネットワークに他の端末が存在しないため，再現することができない．これについても今後の課題としたい．

3. HTTP メッセージフローとログの提示機能

調査の場面において，通信パケットの記録から Web を介した攻撃の疑いがある箇所の抽出作業の過程で，再現した当時の Web サイトの挙動からの候補の判断を容易にするために，HTTP メッセージフローの自動再現と同時にログの取得を行い，収集できたログと HTTP メッセージフローの提示機能について検討する．この機能は，HTTP メッセージフローのステップ再現機能と同時にロガーの起動を行い，再現時に発生する事柄によって生成されるログの取得を行う．その後，収集できたログと再現を行った HTTP メッセージフローを同時に提示する．

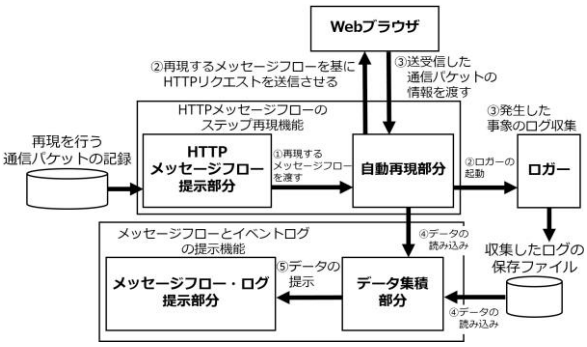

図 3　HTTP メッセージフローの自動再現とログの提示の構成と流れ

Figure 3　Components and flow of automatic reproduction of HTTP message flow and display of logs

当時発生した事柄のログの再取得は，ステップ再現機能が起動するとともにロガーを自動で起動することによって行う．使用するロガーは，外部ファイルにロガーの名前と起動に必要なコマンドをセットにして記載することによって利用者の必要なもののみを選ぶことができるようになっており，追加・削除共に容易に行うことができる．起動したロガーで収集したログは個別に保存し管理する．

HTTP メッセージフローと再収集したログを同時に提示する機能は，図 3 のように，データ集積部分と HTTP メッセージフロー・ログ提示部分から構成されており，再現によって発生した事柄によるログの収集と提示を行う．

データ集積部分は，HTTP メッセージフローのステップ再現機能で自動起動したロガーで収集したログの保存ファイルと，自動再現によって発生した通信パケットのフローデータを読み込み，提示のために表形式で保持，管理を行う．HTTP メッセージフロー・ログ提示部分は，データ集積部分で管理しているログデータとフローデータの表を基に，利用者に対し GUI を提供する．ログデータの提供において，項目ごとにソート処理やフィルタリングを行うことができる．表示する項目はフィルタによる指定がない場合は，表示するログファイルの内容に準拠する．Process Monitor[5]の場合，フィルタ設定なしでは，Time，Process Name，PID，Operation，Path，Result，Detail の 7 項目が表示される．

HTTP メッセージフローと収集したログの提示の流れを以下に示す．

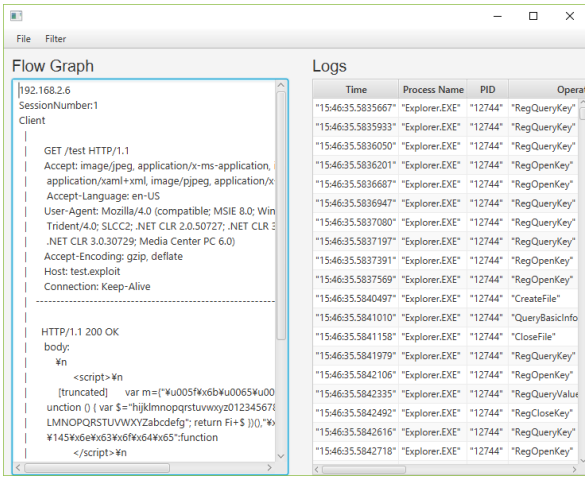

図 4　HTTP メッセージフローと収集したログの提示画面

Figure 4　Display of HTTP message flow and collected logs

① HTTP メッセージフローのステップ再現機能を用いて調査対象のフローの自動再現を行う
② 同時に起動されるロガーを用いてログの収集を行い，
③ 収集できたログを保存する
④ 自動再現によって発生したフローデータと保存されたログデータをデータ集積部分が読み込む
⑤ フィルタ情報を基に表示内容を確定する
⑥ HTTP メッセージフローと収集したログを出力する

HTTP メッセージフローとログの画面出力を図 4 に示す．ウィンドウは左右に大きく 2 つに分かれている．左側には調査対象である通信パケットの記録から抽出した HTTP メッセージフローが表示される．ここでは HTTP メッセージフローのステップ再現機能における HTTP メッセージフロー提示画面と同様に，通信の向きや通信パケットの概要，セッションの通し番号が表示される．通信パケットの概要として，リクエスト方向の通信パケットではリクエストラインと，User-Agent，Host，Cookie の情報がデフォルトで表示される．レスポンス方向の通信パケットには，ステータスライン，Content-Type，Set-Cookie のほか，ペイロードの一部が表示される．リクエストラインとステータスライン，Content-Type については，マルウェア通信の実態調査[6]において，HTTP 通信の調査項目として挙げられているため，Web を介した攻撃による通信の判別手段として表示している．User-Agent 情報は Web クライアントの判別に用い，Host 情報は，特徴的な URL へのアクセスを確認する際に用いる．表示する情報は設定によって変更することができ，現在デフォルトで表示している情報以外にも増やすことや，反対に現在表示している内容から減らすことも可能である．レスポンスのペイロード部分については，表示する分量の調節も可能になっている．右側には，収集したログのエントリを表示することができる．表示するログは，対応可能なロガーのリストから表示するログを選択することで選択することができる．図 4 で提示している内容は Process Monitor で収集したプロセスの記録を読み込み表示した例になる．この例では，フィルタ設定は何もしていないため，すべて表示されている．フィルタ設定を行うことで特定の事柄に関連した内容のみ表示することもでき，また，不要な情報については表示項目ごと非表示にすることもできる．

左右の画面にあるスクロールバーは時間情報で同期している．例えば左側の HTTP メッセージフローを提示する画面でスクロールをすると，スクロール後の HTTP メッセージフローの最も上にある通信パケットの時刻に合わせて右側のログを提示する画面も同時刻のエントリまでスクロールされる．これにより通信が行われたときどのような事柄が発生していたか時系列の把握が容易になると考える．

これらの機能を PC(CPU: Intel core i7 3.3GHz, OS: Windows10 Pro 64bit, Main memory: 32GB)上に試作した．この機能は Java（JRE: build 1.8.0_181-b13, JVM: build 25.181-b13, mixed mode）を用いて試作した．

HTTP メッセージフローの自動再現とログの提示機能によって，調査対象の通信パケットの記録に含まれる HTTP メッセージフローと，再現によって発生した事柄のログを同時に提示し，何時どんな通信が行われて，その結果どのようなことが起きたのかを知ることで，調査の場面においてより詳細な調査を行うべき対象を抽出できる．

4. 実験

本稿で試作した HTTP メッセージフローとログの提示機能について，Web を介した攻撃を見つける調査に試用することで，再現の開始と同時にロガーが起動されることと，読み込んだログファイルの内容が過不足なく表示されること，読み込んだログファイルに対してフィルタリングやソートが正しく実行されること，スクロールバーの同期が正しく行われていることを確認するために実験を行った．また，通信パケットの記録から Web を介した攻撃に該当するところを抽出する過程を支援できることを，Web を介した攻撃を見つけられた数と調査に要した時間を計測し，HTTP メッセージフローのステップ再現機能のみを用いる場合と比較することによって確認する．

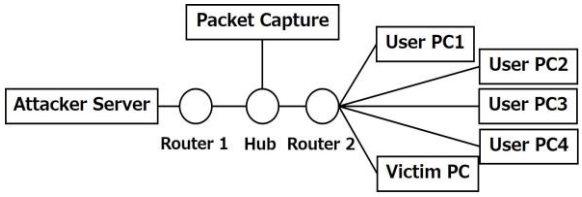

図 5　実験用ネットワークのトポロジ

Figure 5　Topology of the experimental network

実験で用意したネットワークを図 5 に示す．攻撃者の用意

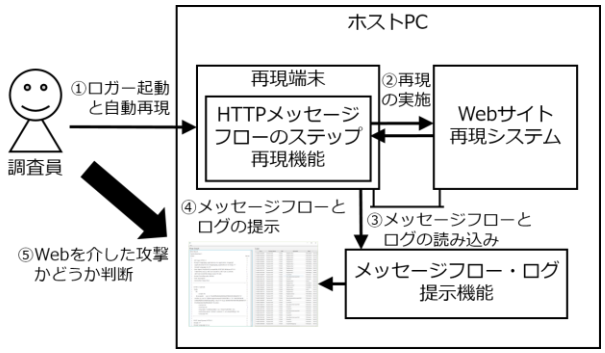

図 6　試作した機能を用いた再現の流れ

Figure 6　Flow of reproduction using prototype function

した Web サイトをホストする Attacker Server (OS: Kali Linux, カーネル: Debian 4.9.30-1kali1)が 1 台と，普段と同様の使い方をしてもらう User PC (OS: Windows10 64bit)が 4 台，調査対象となる Web を介した攻撃の被害を受ける Victim PC (OS: Windows 7 32bit SP1 English, Web クライアント: Internet Explorer 8)が 1 台，発生する通信パケットを採取・記録するための Packet Capture 用 PC(OS: Windows10)が 1 台の計 7 台を用いてネットワークを構成した．また，通信パケットの記録の中から Web を介した攻撃を見つける調査に用いる PC(OS: Window10 Pro 64bit, CPU: Intel Core i7 3.3GHz, Memory: 32GB)を 1 台用意した．

この環境を用いて，Wireshark[7]で通信パケットの収集を行い，収集した通信パケットの記録を用いて，通信パケットの記録の中から Web を介した攻撃を見つける調査を著者が HTTP メッセージフローのステップ再現機能のみを用いる場合と，試作した機能も用いる場合の 2 パターン行った．HTTP メッセージフローのステップ再現機能のみを用いた場合では，調査対象の通信パケットの記録を読み込み，提示された HTTP メッセージフローから，特徴的な通信が行われているフローを探しその HTTP メッセージフローの再現を行い，再現端末上で起動している Process Monitor に表示される，再現時に発生するプロセスを基に Web を介した攻撃を探した．試作した機能を用いて行った調査の手順を図 6 に示す．まず，調査対象の通信パケットの記録を読み込み，提示された HTTP メッセージフローから，特徴的な通信が行われているフローを探し，その HTTP メッセージフローの再現を行う．その後，再現端末上で起動している Process Monitor で収集した，再現時に発生するプロセスを記録したログファイルを試作したシステムを用いて，HTTP メッセージフローとそれによって発生したプロセスを基に Web を介した攻撃を探した．再現に用いる環境として，PC(CPU: Intel core i7 3.3GHz, OS:Windows10 Pro 64bit, Main memory:32GB)上に Virtual Box を使用して再現端末(OS:Windows7 32bit SP1 English, Web ブラウザ:Internet Explorer 8)と Web を介した攻撃の振る舞いを再現するシステム用のホスト (OS: Ubuntu 14.04 LTS)を配置し閉じたネットワークで接続した．通信パケットの記録の中から探し出す Web を介した攻撃として，Metasploit[8]を用いて，MS14-064 OLE 脆弱性を利用しユーザの使用する端末にバックドアプログラムを設置する Drive-by Download 攻撃を Attacker Server に用意した．この Drive-by Download 攻撃への誘導方法として，JavaScript を用いるもの，iframe タグを用いるもの，Clickjack 攻撃を用いるもの 3 種類を用意した．

実験に用いる通信パケットの記録を収集するために，著者を除く当研究室の学生 5 人に協力してもらった．5 人には，それぞれ User PC と Victim PC を使用して論文調査を行ってもらった．このうち，Victim PC の使用者には通信パケット採取期間中に好きなタイミングで Attacker Server がホストする攻撃用サイトにアクセスしてもらった．パケット採取時間は 2 時間とし，通信パケットの採取を行った結果，32,163 セッションが収集できた．収集した通信パケットの記録を用いて，著者が調査を行った．その結果，HTTP メッセージフローのステップ再現機能のみを用いたパターンでは調査に 39 分かかり，試作した機能も使用するパターンでは 24 分かかった．2 パターンとも Web を介した攻撃をすべて通信パケットの記録から探し出すことができた．また，試作した機能が正しく動作することを確認した．

図 5 に示す実験用ネットワークにおいて Router1 を LAN と WAN の境界と見立て，ゲートウェイ部分でのパケット収集を行った．収集した通信パケットの記録を基に再現し，Web を介した攻撃の調査を行った．調査で行った再現作業において，使用した再現端末は 1 台となっている．現時点でのシステム利用において使用する再現端末は 1 台を考えており，複数端末を同時に多数展開し自動で一斉に再現を実施することに関しては，再現環境の自動構築も含めて今後検討していきたい．

本システムを使用するにあたり，通信パケットの記録を収集・保全するために必要となる設備について筆者の所属する研究室を例に検討する．筆者の所属する研究室で使用する端末の総数は 20 台で，参考として 1 ヵ月の総通信量は約 90GB であった．このことから 1 年間の通信量は 1080GB と予想される．通信パケットを取得，タイムスタンプやデジタル署名を計算，付加するデータ保全機能を有するスニッファの機器と，1500GB 程度の容量を持った記憶装置が必要と考えられる．

5. 関連技術

著者らの開発してきたシステムでは，通信パケットの記録から Web サイトとの通信を抽出するにあたり，バイナリデータからパケットの再構築を行っている．同様の機能を持つツールとして，Wireshark[7]や CapAnalysis[9]などがある．それぞれのツールは，PCAP ファイルなどのバイナリファイルを読み込み，パケットごとに切り分けたうえでプロトコル解析を行い，それぞれ可視化する．本研究で開発

したシステムは，Web を介した攻撃を再現するために，通信パケットの記録存在するパケットをネットワークに送信する．

パケットの送信機能を持つツールとして，tcpreplay[10]や，tcpliveplay[11]が存在する．tcpreplay は，元々，IDS や IPS に対して悪意あるトラフィックを送信しその性能を検証する目的で作成されたパケット送信ツールである．現在は Tcpreplay というオープンソースユーティリティ群のツールの 1 つとなっており，任意のスピードでパケットを送出することができるほか，送出するパケットの動的編集などが行える．その他，PCAP ファイルを直接編集するなどの機能も存在している．しかし，tcpreplay は純粋にパケットを送出する機能しか有さず，HTTP 通信に必要となる TCP コネクションの確立などを行うことができない．よって，本研究が対象とする Web を介した攻撃の再現に用いることはできない．tcpliveplay は，動的脆弱性テストを実施するために開発されたツールである．tcpreplay と同様パケット送信機能を備え，そのうえで TCP コネクションを考慮したパケット送出が可能となっている．tcpliveplay が再生に使用できる通信パケットの記録は単一の TCP フローのみである必要がある．本研究で構築したシステムは与えられた通信パケットの記録から TCP フローを抽出し，それぞれ個別に管理しているため，受信したリクエストに対し，適切なレスポンスを返送することができる．

Web ページの再現ができる機能を持つ製品として NIKSUN 社製の NetDetector[12]と NetVCR[13]がある．これらの製品は，ネットワーク内で行われる通信をキャプチャし，フォレンジック分析に向いた形式にまとめて提供することができる．これらの製品の機能として，キャプチャした，もしくは保存されたパケットのデータから Web アクセスのセッションを復元し，当時アクセスが行われた Web ページを表示するものがある．本研究で開発したシステムでは復元した Web サイトに改めてアクセスすることで当時アクセスした Web ページとアクセスしたときの挙動を再観測できる．

社内ネットワークなどの環境を再現し，その中で実際のマルウェアや攻撃などを発生させ，その挙動をホストおよびネットワークの両面から分析し，対処方法が体験できる小規模攻撃再現環境の技術が開発されている[14]. また，ソーシャルエンジニアリングなどにより，あらかじめ入手した攻撃対象の環境情報を利用したマルウェアが増加している．そのため，動的解析を行うには，攻撃対象を含む環境情報も再現しないと，マルウェアの活動の全体が把握できないことから，高詳細な攻撃対象環境を模倣した観光の構築と動的解析を自動化するシステムが提案されている[15]. 著者らの開発してきたシステムは，インシデント対応の初動対応や調査の場面での使用を想定しており，入手できた痕跡の情報（通信パケットの記録）から，当時行われた攻撃を再現することで，インシデントに関する事象や OS やアプリケーションの振る舞いの履歴などで得られなかったものを収集し，解析を支援することを目的としている．

マルウェアが設置，実行され権限昇格を経てサーバに侵入，Web 上に資料等が流出したというインシデントを想定したときの，本システムの利用例を考える．このようなインシデントが発生した場合，以下のようなインシデント対応が考えられる[16].

1. Web から資料が閲覧できない様にする
2. ログなどから漏洩元のサーバなどへのアクセス候補を抽出
3. 候補からマルウェア設置，実行などの原因の調査

上の 3.の作業において，Web を介した攻撃によってマルウェアが設置された場合，プロキシや IDS・IPS のログなどから攻撃の全体像を把握する[17]．攻撃の全体像を把握すると，感染した端末を起点に，感染経路や利用された脆弱性，マルウェアとそれによる影響の範囲について調査を行う. このとき, 本システムを用いて攻撃を再現することで，Exploit が受理されたかを確認することができ，これにより，感染経路や感染端末の調査が行える．さらに，設置されたマルウェアの動作を観測し，マルウェア本体を回収することにも利用できると考えられる．

6. まとめ

これまで著者らは，インシデント対応における調査活動を支援するためのフォレンジック支援システムを開発してきた．本稿では，通信パケットの記録からセッションを抽出し，その HTTP メッセージフローの自動再現を行うと同時に，再現端末上で OS やアプリケーションなどのログを再取得する機能について検討を行った．

実験では通信パケットの記録の中から Web を介した攻撃を見つける調査を試作した機能を用いて実施し，調査対象の Web を介した攻撃が発見できることと，試作した機能が設計した通りに動作することを確認した．

著者らのシステムは，HTTPS の通信パケットの記録から暗号化されたリスエスト，レスポンスを抽出して Web サイトを復元するといったことに対応していないが，これについては今後の課題としたい．

参考文献

[1] Jason T. Luttgens, Matthew Pepe, Kevin Mandia：Incident Response & Computer Forensics, Third Edition, NIKKEI BP. INC. (April 2016).
[2] 奥田裕樹, 福田洋治, 白石義明, 井口信和:ドライブ・バイ・ダウンロード攻撃によるインシデントを再現するフォレンジック支援システム, 電子情報通信学会技術研究報告(ICSS), Vol.117, No.125, pp.81-86(2017).
[3] Selenium Project: Selenium – Web Browser Automation, available from< https://www.seleniumhq.org/> (accessed 2018-07-24).
[4] 岩崎正治，原田要之助：Advanced Persistent Threat(APT)の分類と対応策についての一考察，システム監査学会，Vol.26,

No. 2, pp.2-15(2013).
[5] Microsoft: Process Monitor, available from<https://technet.microsoft.com/ja-jp/sysinternals/processmonitor.aspx>, (accessed 2018-09-10).
[6] 畑田充弘, 森達也：未知マルウェア検知に向けたマルウェア通信の実態調査, コンピュータセキュリティシンポジウム 2015 論文集, Vol.2015, No.3, pp.520-527(2015),
[7] Wireshark Foundation: Wireshark. Go Deep., available from<https://www.wireshark.org/>, (accessed 2017-06-15).
[8] RAPID7 metasploit: Metasploit Testing Software, Pen Testing Security, available from<https://www.metasploit.com/>, (accessed 2018-09-10).
[9] Gianluca Costa: CapAnalysis, available from<https://www.capanalysis.net/ca/>, (accessed 2018-05-23).
[10] AppNeta: Tcpreplay – Pcap editing and replaying utilities, available from<https://tcpreplay.appneta.com/>, (accessed 2018-05-23).
[11] AppNeta: tcpliveplay man page, available from<http://tcpreplay.appneta.com/wiki/tcpliveplay-man.html>, (accessed 2018-05-23).
[12] NIKSUN Incorporated.: NIKSUN | NetDetector, available from < https://www.niksun.co.jp/products/netdetector.html>, (accessed 2018-11-08).
[13] NIKSUN Incorporated.: NIKSUN | NetVCR, available from < https://www.niksun.co.jp/products/netvcr.html>, (accessed 2018-11-08).
[14] 三輪 信介, 門林 雄基, 篠田陽一：小規模攻撃再現テストベッドによる動作記録データセットの生成, マルウェア対策研究人材育成ワークショップ(MWS2009)発表資料, A9-2(2009).
[15] 安田真悟, 三浦良介, 高野祐輝, 宮地利幸：Alfons：マルウェア解析の為の高詳細な環境構築システム, 電子情報通信学会技術研究報告, ICM, 114(523), pp.139-144(2015).
[16] Information-technology Promotion Agency, Japan（IPA）：情報漏洩対策のしおり（第 7 版）, 入手先＜ https://www.ipa.go.jp/security/antivirus/shiori.html#roei＞ （参照 2018-06-11）.
[17] 折原慎吾, 鐘本楊, 神谷和憲, 松橋亜希子, 阿部慎司, 永井信弘, 羽田大樹, 朝倉浩志, 田辺英昭：セキュリティのためのログ分析入門, 技術評論社（2018）.