G-24

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ステップ再現の検討—

Reproduction of Attacks via Web by Using Captured Packets -A Function Performing Phased Reproduction of HTTP Message Flow-

奥田 裕樹†　福田 洋治‡　井口 信和‡
Yuki Okuda　Youji Fukuta　Nobukazu Iguchi

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正確に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，そのフローの自動再現を行う HTTP メッセージフローのステップ再現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自動で再現すると同時に，送受信された通信パケットが再現するメッセージフローのものと一致するかチェックを行う．再現するメッセージフローに含まれるリクエストが送信されない場合は，自動で次のリクエストを送信する．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挙動を再現することができる．

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

このシステムは，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 サイトに対し，仮想環境上の Web ブラウザからアクセスすることでアクセス時の Web ブラウザの挙動がそのまま再現される[2].

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

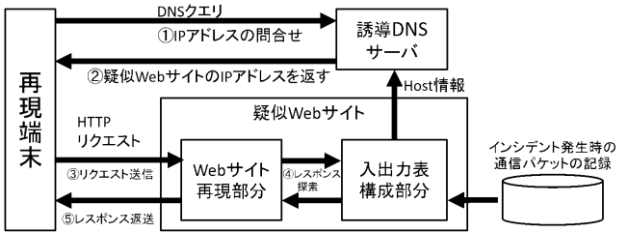

図 1 Web を介した攻撃の振る舞いを再現するシステムの構成と動作

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 ブラウザに対し返送する
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

以上の流れで Web ページを復元する．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 ページとその挙動が再現できる.

再現端末は仮想化技術を用いて被害の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感染し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.

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で確認した[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．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.

†近畿大学大学院 総合理工学研究科，Graduate School of Science and Engineering Research, Kindai University

‡ 近畿大学理工学部，Faculty of Science and Engineering, Kindai University

3. HTTP メッセージフローのステップ再現

これまで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セッションを抽出し，当時と同じ HTTP リクエストに対しそれに対応するレスポンスを返すことで，当時の Web サイトの挙動を再現することができる．通信パケットの記録から Web サイトを復元することにより，調査の場面で当時と同一の環境からアクセスすることで当時の攻撃が再現でき，実際に発生した事柄が観測できる．

本稿では，調査の場面において，当時の Web サイトの挙動を再現する作業を容易にするために，HTTP メッセージフローの自動再現を行う，メッセージフローのステップ再現機能について検討する．この機能は，再現を行う通信パケットの記録からセッションを抽出し，利用者にそのフローの様子を提示する．その後，利用者が再現を行う HTTP メッセージフローを選択すると，Web ブラウザを起動しその Web ブラウザを用いてそのフローの再現を行う．

本機能は図 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 ブラウザ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 メッセージフローを利用者に提示する．HTTP メッセージフロー提示の様子を図 3 に示す．HTTP リクエストの通信パケットは，Client から Server への矢印の上にリクエストラインとヘッダ情報が記載されている．レスポンスの通信パケットは Server から Client への矢印の上にステータスラインとペイロードの中身が記載されている．この提示した HTTP メッセージフローを基に再現を行う 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 ブラウザを操作して再現させる．自動再現部分では，Web ブラウザ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 ブラウザ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 ブラウザを操作して HTTP リクエストを送信する
② Web ブラウザ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 ブラウザを操作し 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 メッセージフローと共に提示する．

HTTP メッセージフローのステップ再現機能によって，再現対象となる通信パケットの記録に含まれる HTTP メッセージフローを自動的に再現することができるようになり，調査の場面において，Web を介した攻撃を再現して，起こった事柄を観測しその挙動を把握する調査が，アクセス操作を自動化できることによって容易に行えるようになると考えられる．

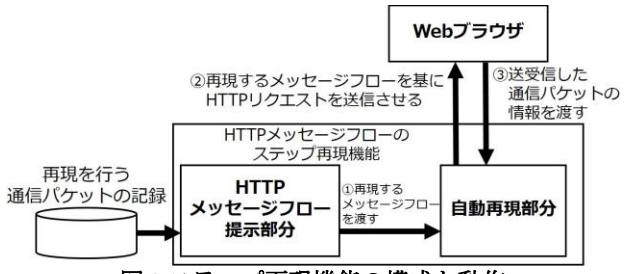

図 2 ステップ再現機能の構成と動作

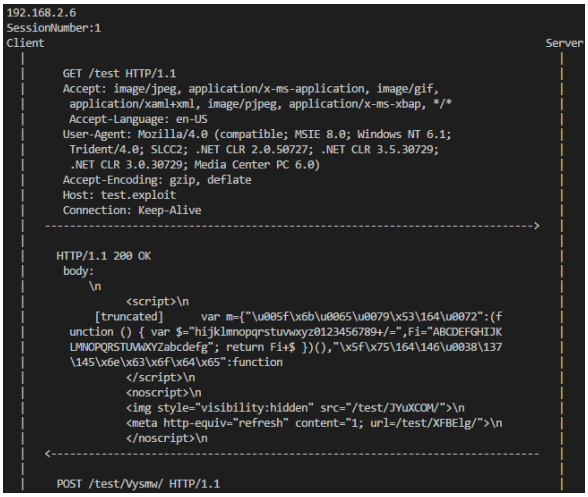

図 3 メッセージフロー提示画面

4. 実験

HTTP メッセージフローのステップ再現機能を用いることによって，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が容易になることを，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，ネットワークフォレンジックツールを用いる場合と比較することによって確認する．比較対象として，Web セッションを分析するツールである， Wireshark(Ver. 2.6.2)[4] と Network Miner(Ver. 2.3.1, free 版)[5]を用いる．

実験で用意したネットワークを図 4 に示す．攻撃者の用意した 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 ブラウザ: 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)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，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，ネットワークフォレンジックツールを用いる場合とステップ再現機能を用いる場合の 2 パターン行った．ネットワークフォレンジックツールを用いる場合では，Wireshark を用いて通信の様子を確認し，複数回のリダイレクトや特徴的な URL へ

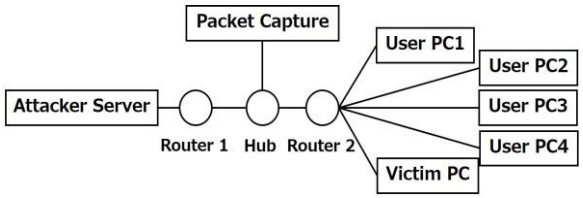

図 4 実験用ネットワークのトポロジ

のアクセスなど，Web を介した攻撃の可能性のある候補を発見したあと，Network Miner を用いてダウンロードされたファイルの確認を行うという流れで Web を介した攻撃の可能性がある候補を挙げる調査を行った．HTTP メッセージフローのステップ再現機能を用いた場合では，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．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[6]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させる 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，著者がそれぞれの調査を実施した結果，ネットワークフォレンジックツールを用いる場合には 139 分かかり，ステップ再現機能を用いた場合には 53 分かかった．また，両方のパターンでの調査において，Victim PC が受けた Web を介した攻撃による通信をすべて発見することができた．

ネットワークフォレンジックツールを用いた調査では，あくまで特徴的な URL や特徴的なリダイレクトのパターンを基にダウンロードされたファイルを確認することで，Web を介した攻撃の候補を探すことまでしかできなかった．著者らがこれまで開発してきた Web を介した攻撃の振る舞いを再現するシステムと HTTP メッセージフローのステップ再現機能を組み合わせて用いた場合では，再現を行うという点から，当時実際に行われた Web を介した攻撃の挙動を観測することや，Process Monitor のプロセスの記録から Exploit を受理してブラウザから別のプロセスが実行されていることが確認できることから，実際に攻撃が行われたことまで確認することができる．また，調査に要した時間についても HTTP メッセージフローのステップ再現機能を用いた方が 86 分短縮できたことから，HTTP メッセージフローのステップ再現機能を用いることで Web を介した攻撃の調査を容易に行うことができるようになると考える．

5. 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，メッセージフロー単位での自動再現を行う HTTP メッセージフローのステップ再現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 HTTP メッセージフローのステップ再現機能用いる場合とネットワークフォレンジックツールを用いる場合とで実施し，Web を介した攻撃に用いられた通信を見つけられた数と調査に要した時間を比較した．

ネットワークフォレンジックツールを用いる場合と比較して，再現することで Web を介した攻撃の挙動を観測できることや，ブラウザから別のプロセスが発生することを確認できることで候補ではなく実際に攻撃が行われた通信を見つけることができる点と，調査に要する時間が短縮できた点から，HTTP メッセージフローのステップ再現機能によっ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が容易になると考えられる．

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] Wireshark Foundation: Wireshark. Go Deep., available from<https://www.wireshark.org/>, (accessed 2017-06-15).

[5] NETRESEC: Network Miner – The SNM and Network Forensics Analysis Tool, available from<https://www netresec.com/index.ashx?page=NetworkMiner>, (accessed 2018-07-25).

[6] David K, Jim O'G, Devon K, Mati A：実践 Metasploit，オライリージャパン(2012).