HITACHI
Inspire the Next

Hitachi Systems Security Journal

VOL.56

株式会社 日立システムズ

Human * IT

Hitachi Systems security Journal VOL.56

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

サイバー空間の変化を30年以上も見続けてきた研究者が予測するAI時代のセキュリティとは？

Mikko Hypponen ミッコ・ヒッポネン インタビュー

今回、インタビューするのは、世界でもっともその名を知られたサイバーセキュリティの専門家の 1 人であるミッコ・ヒッポネン氏だ。本年 11 月に東京で開催された CODE BLUE 2023 では基調講演に登壇している。「スマートであれば、ぜい弱である」と題された講演は、1990 年代初頭から現在にいたるまで、常に最前線でサイバー空間を見続けてきたヒッポネン氏の 30 年間以上にもわたるキャリアの集大成であり、昨年上梓した著作の書名（邦訳は異なる）にもなっている。「歴史は繰り返す」といわれるように、歴史を学ぶことは未来を予測する力を養うことにもつながる。セキュリティ業界のさまざまなトピックについてヒッポネン氏に話を伺った。

取材・文・撮影 = 斉藤健一
通訳 = 高間剛典

サイバー空間の歴史を知ることは未来を見通す力にもなる

斉藤（以下 S ）：早速ですが、「If It's Smart, It's Vulnerable（スマートならばぜい弱である）」が上梓されました。1990 年代初頭から現在に至るまでのインターネットの進化と、それに伴うサイバー犯罪の手法の変化などがさまざまな角度から述べられています。まさに、ミッコさんのこれまでのキャリアを集大成したものと言えるでしょう。この著作を執筆するにいたった理由を教えてください。

ミッコ・ヒッポネン（以下 M ）：過去 30 年間でサイバー空間は大きく変化しました。そして、この間に起きたことを書き残せる人間は私以外にはいないと考えたからです。私はサイバー空間が登場した初期からセキュリティに携わっていますので、書き残す責任があると感じたのです。このプロジェクトが始まったのは 2011 年のことです。しかし、何年もの間、執筆は進まずにいました。新型コロナウイルスのパンデミックの影響で、執筆の時間を確保することができ、ようやく書き終えることができました。また、この本が日本語を含む 5 つの言語で出版され、世界中の多くの読者に読まれていることを大変嬉しく思います。

S 過去の歴史を知ることは、これからの未来を予測するうえで、非常に重要だと思うのですが、いかがでしょうか。

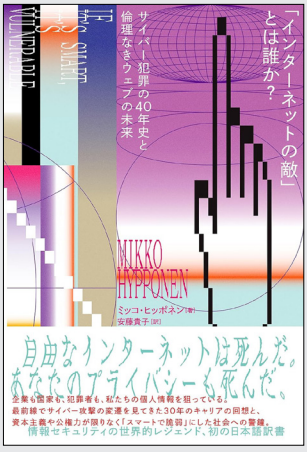
M そのとおりです。いくつかの攻撃手法は何度も繰り返し登場しています。それらはわれわれがすでに解決したと考えていた問題です。その典型例がマクロウイルスと Telnet への攻撃です。マクロウイルスは約 15 年前に駆逐されたと思われていましたが、再流行の兆しが見られます。また、Telnet は暗号化されていないため安全ではないプロトコルです。特に IoT 分野ではまだ使われているケースが多く、Telnet への攻撃が増加しています。

S ここ数年間におけるサイバー空間の変化の速さはどう思われますか。速いと感じますか、それとも想定の範囲内でしょうか。

M どこへ向かうのかを見いだすことは比較的容易ですが、その方向に進む時期を予想することは難しいと思います。AI や量子コンピューターを例に挙げると、これらは過去 15 年以上にわたって議論されてきました。どちらの技術も、議論が盛り上がる時期と、期待が失望に変わり失速する時期がありました。しかし、AI に関しては、ここ 2 年間でかつてないほどの盛り上がりを見せています。量子コンピューターに関しては、過去数十年にわたって開発が進められていますが、いまだ広く実用化されるところにまではいたっていません。最近、この分野で興味深い動きがあるようですが、最終的にどの程度普及するかは不明です。ただ、量子コンピューターと機械学習が組み合わされたときにどのようなブレイクスルーが起こるか注目したいと考えています。

「インターネットの敵」とは誰か？
サイバー犯罪の 40 年史と倫理なきウェブの未来
ミッコ・ヒッポネン著／安藤貴子訳、3080 円（税込）、双葉社

原題の “If It's Smart, It's Vulnerable（スマートならばぜい弱である）” とは、インターネット接続が可能な高機能デバイスほどセキュリティリスクが高いことを指摘するもので、ミッコ・ヒッポネンの法則と呼ばれている。本書は、黎明期から現在にいたるまでのインターネットやコンピューターセキュリティの歴史をさまざまな視点から考察する。その内容は、マルウェアの歴史にはじまり、スマート社会の落とし穴、暗号通貨時代のサイバー犯罪、国家による諜報や情報戦などで、さらに AI がもたらす今後の変化にまで言及している。各章は独立しておりヒッポネン氏の語り口も軽妙だ。興味のあるところから気軽に読むことができるだろう。



S こうした技術はセキュリティ上の脅威にもなり得ますよね。

M そのとおりです。AI を使うことで、世界中のほぼすべての言語で大規模な詐欺を実行することが可能になります。また、AI によりマルウェアのコードを書き換えたり、映像や音声のディープフェイクも簡単に作成できるようになったりします。さらに、これらのマルウェアやディープフェイクを使って、完全に自動化された攻撃キャンペーンを展開することも可能になるでしょう。

S はい。多くのセキュリティ業界関係者の方々が同様の指摘をされています。

M 量子コンピューターでは、使用できる量子ビットが十分に増えると、既存の暗号化アルゴリズムの多くが現実的な時間内に解読可能になるでしょう。すでに量子耐性のある暗号化アルゴリズムが開発されていますが、これを広範囲に実装・移行していくには多くの時間を要すると考えられます。

S 移行に多大な時間を要する理由は何でしょうか。

M Web ブラウザー、サーバー、コンピューター、スマートフォン、自動車など、ありとあらゆるものをアップデートしなくてはならないからです。過去の事例ですが、IPv4 から IPv6 への移行は 25 年も前から始まっていますが、いまだに完全な移行はできていません。量子耐性のある暗号化アルゴリズムへの移行については、IPv4 から IPv6 への移行よりもはるかに時間がかかるだろうと考えています。

S IPv6 の場合にも当てはまると思いますが、移行が加速するようなきっかけは何か考えられますか。

M 実際に何か大きな問題が起きるまで、変化はないと思います。過去のパーソナルファイアウォールの例が参考になると思います。そのアイデアは良かったものの、初期にはほとんど注目されていませんでした。2000 年代初頭、Code Red ※1 や Nimda ※2 などのインターネットワームが流行したことで、導入が急速に進みました。このように問題が起きるまで利用されなかったソリューションの例は多々あります。

ミッコ・ヒッポネン（Mikko Hypponen）
フィンランドのコンピューターセキュリティ専門家、講演者、作家。WithSecure（旧社名 F-Secure）の CRO（Chief Research Officer）を務める。1990 年代初頭から現在にいたるまでセキュリティ業界の最前線で活躍を続けている。

S 現在、ランサムウェアによる被害が世界中で増大しています。このランサムウェアの流行が、以前の Code Red や Nimda のように、その後の社会に大きな影響を与えるとお考えですか。

M われわれは、ランサムウェアがどのような影響を及ぼすかをすでに目の当たりにしています。米国最大の石油ガスパイプライン企業がランサムウェアの攻撃によって操業を停止したことも、世界的なロジスティック企業や世界最大級の食肉加工企業がランサムウェアの被害にあったことも知っているのです。もし、このような大事件で、何か世の中が変わるような動きがあるとすれば、われわれはもうすでにそれを目撃にしていると思います。

S 世界有数の大企業が数多く被害に遭っています

※1 **Code Red（コードレッド）**：※1 Code Red（コードレッド）：2001 年 7 月に大流行したネットワークワーム。Microsoft の IIS サーバーのぜい弱性を利用して感染を広げた。ワームを発見した研究者が清涼飲料水の名前から命名した。2001 年 8 月には Code Red II が登場するが、プログラムとしては Code Red とは全くの別物となっている。

※2 **Nimda（ニムダ）**：2001 年 9 月に感染を広げたネットワークワームで、ファイルに感染するコンピューターウイルスでもある。名称は「admin」を逆から綴ったもの。Windows のぜい弱性や Code Red などのワームによるバックドアを利用など、複数の感染方法があり、短期間にインターネットへの感染が広まった。

が、それでも社会が大きく変わるところまでにはいたっていないと見るべきですね。

M サイバー犯罪者グループは、ダーク Web 上でリークサイトを公開しています。そこには身代金を要求されている企業のリストがあり、指定期間内に身代金が支払われなかった場合は、詐取した情報を公開すると脅迫しています。このリストを見ると、世界中でどれだけ多くの企業がランサムウェアに感染し、被害を受けているのかがわかるはずです。

S ランサムウェアに感染した企業の被害額を考えると、身代金を支払うか否かはさておき、営業停止による損害やその後のセキュリティ対策などに多大な費用がかかります。

M そのとおりです。先にセキュリティに投資しておくべきです。被害に遭ってからでは二重に費用がかかります。それは、まるで火災で家が焼失した後で火災保険に入るようなものです。

アンダーグラウンドでの情報収集

S 次にサイバー犯罪者たちが集まるアンダーグラウンドの世界について伺いたいと思います。アンダーグラウンドのフォーラムなどでは、情報を得るためには、情報を提供するギブ・アンド・テイクの原則で成り立っていると聞きます。この世界でどのような手法を用いて情報収集をされているのでしょうか。

M まず、強調しておきたいのは、われわれは違法行為には一切関与していないということです。詳細はお話できませんが、われわれは、サイバー犯罪者たちが利用するフォーラムや Web サイトについて十分に理解しており、適切な場所で耳を澄ませていれば情報が手に入ることを知っています。これらのフォーラムには、Tor の隠しサービスや Telegram・Signal のグループなどが存在します。さらに犯罪者間で機密情報が交換されるフォーラムは招待制でアクセスすることは困難です。

S アンダーグランドを調査されている方に話を伺うと、招待制のフォーラムの話題がよく出てきます。

M われわれは長年、犯罪者が存在する環境でオンラインのアイデンティティをいくつも作り上げてきましたので、これらにアクセスすることが可能です。サイバー犯罪者のフォーラムは、その性格上、閉鎖と開設を繰り返しており、新たなフォーラムが開設される際に、彼らが信頼できると判断したアイデンティティは招待されます。これによって継続的に情報を入手しているわけです。また、このような手法で情報を収集していることは秘密ではありません。われわれの動きをサイバー犯罪者に知らせることで、もしかしたらフォーラム内にスパイがいる可能性を彼らに疑わせ、不安を煽る目的もあるからです。

サイバー空間での攻防の行方

S サイバー空間における攻撃側と防御側との争いでは、攻撃側が圧倒的に有利だと言われます。この構図は今後も変わりないのでしょうか。

M 答えは「イエス」とも「ノー」とも言えます。攻撃側は攻撃のタイミングや手法を自由に選ぶことが可能です。一方で、防御側も長年にわたる機械学習の成果を活用した自動化を進めています。この進展が、今の攻撃側と防御側のパワーバランスに影響を与える可能性があります。

S 防御側の自動化について、もう少し具体的にお聞かせ下さい。

M 機械学習によるアノマリ検知です。組織内ネットワークでの通常の稼働状態を定義し、そこから逸脱する「異常な挙動」を迅速に検出します。例

CODE BLUE 2023 日立システムズのブースにて

えば、「なぜこの時間にこのようなトラフィックが発生するのか」といったことです。この方法は、既知の攻撃はもとより、未知の脅威にも対応可能です。また、この分野における自動化と AI の活用は非常に効果的であると考えています。もちろん、このシステムも完璧ではありません。攻撃でないものを攻撃と判定するフォルス・ポジティブの問題もあります。しかし、現状ではこれが最も効果的なシステムだとわれわれは考えています。

S 確かに。攻撃側がこれを回避しようとするのは、とてもハードルが高くなると思います。

M 話がすこしそれますが、IT セキュリティの成功事例の 1 つとして、2007 年に登場した iPhone を挙げることができます。このデバイスはコンピューターでありながら、その所有者がプログラミングできないという点が、セキュリティを強固なものにしています。

現状において、iPhone や Android は、広く普及したコンピューターであり、ユーザーが直接プログラミングできないため、非常に安全であるといえます。

S なるほど。プログラミングできないデバイスに対して、攻撃側が有利に立つのは難しいですね。

M はい。このことを端的に示す例が「Pegasus」です。イスラエルの企業が開発したモバイル端末用のスパイウェアで、国家レベルで活動家や反体制派、ジャーナリストを監視する目的で使用されていました。なぜ、この事例が重要なのかというと、iPhone1 台に Pegasus を感染させるのに 1 万ドルもの費用が必要だったためです。このため、Pegasus は特定の重要な対象にしか使うことができません。高額な費用をかければ攻撃できるということは、セキュリティの敗北を意味するでしょうか。私はむしろ成功例だと考えます。例えば、Windows ユーザーを 1 人感染させるためにかかる費用と比較すれば、その差は歴然です。

S 攻撃者にとっての費用対効果を見合わないものにするというのもセキュリティ戦略の 1 つですね。非常にユニークな視点だと思いました。

オープンソース AI の功罪

S 昨年の ChatGPT や Stable Duffusion の登場以来、一般の人にとっても AI の分野で起きていることが理解しやすくなりました。現在、オープンソースの AI が開発され、個人のコンピューターにもインストールできるようになっています。攻撃者はこれを利用し、自分たちのニーズに合わせてトレーニングしています。結果として、マルウェアを感染させるためのフェイクメールの作成などに AI が使われ、サイバー攻撃の効率が向上していることが考えられます。こうした状況についてどのようにお考えですか。

M 以前、OpenAI とミーティングをしたことがありますが、彼らは悪意のある使用を常に監視しており、そのようなユーザーを積極的に排除しています。ですが、個々のローカル環境に構築した AI は、その監視の範囲外にあります。私は一貫してオープンソースの支持者でしたが、オープンソースの AI については不安を持っています。

S かつての OpenAI は研究のすべてを公開していましたが現在は違いますね。

M それには 2 つの理由があると思います。1 つはソースコードを持っていれば、そこに組み込まれているセキュリティ機能を削除できるためです。もう 1 つの理由は、現在の国際情勢における大きな対立があることです。この状況下では、OpenAI が提供する重要なツールを、例えばロシアや中国のような権威主義国家にも与えることとなります。さらに、AI は AGI（Artificial General Intelligence：汎用型人工知能）と呼ばれているものにどんどん近づいています。私自身としては、AGI が出てくるときに、西側の民主的でセキュリ

ティ意識が高い企業の一員として関わっていきたいと考えています。

AI 時代のサイバーセキュリティを予測する

S CODE BLUE の基調講演では、今後のセキュリティは「AI vs. AI」になるというお話がありました。これには、どのようなシナリオが考えられるでしょうか。

M 次に予想されるのは、「サイバー犯罪者のユニコーン」（犯罪ビジネスを通じて巨額の富を得ているグループ、ユニコーン企業になぞらえた名称）の出現です。彼らは、現在手動で行なわれる作業を自動化する方向に進むでしょう。例えば、新たなランサムウェアのバージョンを開発する場合、マルウェアをメールに添付して送信するか、有害なリンクを含めたメールを送信してターゲットにマルウェアをダウンロードさせるような方法が考えられます。しかし、これらの手法は自動化された防御システムであれば、検出していくことは可能なわけです。

S 攻撃対象のドメインからブロックされたことで、防御システムに検出されたことに攻撃者も気づくわけですね。

M はい。防御側の自動化システムに対して、攻撃者は新たなターゲットを探し、そのドメインに特化したメールを作成し、ターゲットに合わせてマルウェアも再コンパイルする必要があります。現在、攻撃側がこのような対応をするのに数日かかることが、われわれが行なうネットワーク監視によって明らかになっています。このことは攻撃側の対応が人手によるものであることを示していると思います。攻撃側の自動化については、技術として存在していますが、それを実行しているグループについては、われわれが監視している範囲では、まだありません。そして攻撃側のシステムが自動化されたときに「AI vs. AI」の構図となります。

S お話を伺うと、「AI vs. AI」の時代となるのも間近のように思えます。

M 明日そうなったとしてもまったく不思議ではありません。われわれは 2005 年から機械学習による防御システムの開発に取り組んでいます。いわば、攻撃者に対して 18 年分のアドバンテージがあるわけです。ですから、「AI vs. AI」の時代となっても防御側が勝利すると信じています。ただし、今後どのように状況が変わるかは予測できません。

S 今後のサイバーセキュリティについて視野が広がりました。最後に、これからを担う若いエンジニアや業界をめざす人たちに向け、メッセージをお願いします。

M 将来、セキュリティエンジニアの重要性はこれまで以上に高まるでしょう。技術分野での才能や長年にわたるトレーニングにより磨かれた能力をどう活かすかは非常に重要です。人によって、それらの能力はサーチエンジンや OS やゲーム開発などに向けられますが、セキュリティ業界で働くということは、その才能を他者のために使うことを意味します。犯罪者たちはその能力を悪用して他のユーザーを侵害することを目的としていますが、セキュリティエンジニアは他者を助けることを目的としています。助けを求める人々に支援を提供することは非常に価値ある行為です。

S 本日は本当に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 2023.12

文＝日立システムズ

海底ケーブルを取り巻く脅威と有事への備え

【概要】国際通信の約 9 割は海底ケーブルで接続されており、日本においては約 30 本のケーブルが敷設されている。海底ケーブルは衛星通信と比べてコストが低く、高速で安定した大容量通信が可能である一方、災害や人的要因による物理切断のリスクがある。日本ではこうした非常事態に備えるべく、東京・大阪圏に集中しているデジタルインフラを地方に分散・整備する取り組みを進めている。これらの海底ケーブルや陸上の通信網にケーブルを接続するための陸揚局への安全対策強化と並行して、平時から信頼のおける有志国との相互接続性を強化しておくことが重要である。

【内容】2023 年 8 月、アフリカのコンゴ川河口付近において海底地滑りが発生。欧州と西アフリカをつなぐ 2 本の海底ケーブルが断線し、周辺地域で通信の遅延などの問題が発生した。日本における海底ケーブルの断線事例としては、2011 年の東日本大震災が挙げられる。この震災では、関東近海の海底ケーブル 10 か所が切断され、国内の通信に多大な影響を及ぼした。現在、国際通信の約 9 割は海底ケーブルが占めており、全世界で 400 本以上が敷設されている。そのうち日本近辺の海域では約 30 本のケーブルが国外通信用として各国に接続されている。海底ケーブルが衛星通信と比較して優れている点として、①保守が容易でコストが低いこと。②安定した大容量の通信が可能であること。③物理的な伝送距離が短いことなどが挙げられる。こうした多くのメリットがある一方で、自然災害や人的要因によって物理的に切断されてしまうリスクがある。非常時には一時的な伝送手段としてヘリコプターによる無線通信の中継や、衛星通信が使われることもあるが、通信帯域や伝送距離の課題があり、完全に代替することは現実的ではない。そのため、ケーブルが切れても通信を維持できるように海底ケーブルの複線化や陸揚局の地方分散が重要な課題である。

日本における複線化や地方分散の取り組みも進められており、デジタルインフラ（DC など）整備に関する有識者会合によると、現在東京・大阪圏に集中しているデジタルインフラについて、北海道や九州エリアにおいても整備を促進する方針だという。この背景として、北米やアジア太平洋をつなぐ地理的優位性を活かした国際的なデータ流通のハブとしての機能を強化することや、大規模自然災害への備えとしてのレジリエンス強化が挙げられている。しかしながら、海底ケーブル切断の脅威は災害だけでなく、サイバー攻撃や電力供給の途絶、磁気嵐などさまざまな要因が存在する。

昨今のウクライナ侵攻や台湾有事のリスクから安全保障上の懸念も高まっており、船舶や水中ドローンによる人為的破壊、光信号増幅器への盗聴器設置のほか、陸揚局が公開情報から特定可能などがリスクとして考えられ、事業者と政府が適切に連携できる体制の構築や、ケーブル敷設・修理船への支援など、海底ケーブルや陸揚局の安全対策強化も重要な課題である。加えて、有事の際には官民を問わず重要なデータの保全を担保するための対応も求められる。他国の例としては、ウクライナ政府はロシアによる侵攻を受けて政府が保有する重要データを国外のデータセンターへ移行したとされており、日本においても安全保障環境の変化に応じた対応の備えが必要である。平時から信頼できる有志国との相互接続性を強化しておくことや、Starlink などの海底ケーブルの切断に備えた通信経路を確保しておくことが重要である。

［情報源］https://www.meti.go.jp/policy/mono_info_service/joho/conference/digital_infrastructure.html

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 IoT 検索エンジン！

1. 基礎知識編

文＝日立システムズ

1. はじめに

各種セキュリティツールを実践的に紹介する連載企画、「レッツトライツール」が Vol.50 よりスタートしました。第一弾では「HDD 保全」、第二弾では「Windows システム確認」を 3 回に分け、各工程を紹介してきました。今回からは「IoT 機器探索」と題して「Shodan（ショーダン）、Censys（センシス）」といった IoT 検索エンジンを用いたぜい弱性確認手法などを解説します。自組織が管理しているサーバーが外部からどのように見えているのかといった確認に利用したり、管理しきれていない隠れたサーバーなどを探索したりし、リスクの軽減に活用可能です。

「IoT 検索エンジン」は次の 4 部構成となっています。

1. 基礎知識編
   Nmap を利用して、ポートスキャンを試行します。
2. 所有サーバー確認編
   自身が管理している IP アドレスなどがわかるサーバーが「Shodan、Censys」といった IoT 検索エンジンでどのように見えるのかを確認します。
3. サービス探索編
   「Shodan、Censys」といった IoT 検索エンジンを用いて、探索したいサービスが稼働しているサーバを確認します。自組織で管理できていないサーバを探索する際に利用します。
4. サーバー探索編
   「Shodan、Censys」といった IoT 検索エンジンを用いて、サーバーを探索します。自組織で管理できていないぜい弱なサーバを探索する際等に利用します。

IoT 検索エンジンと呼ばれる「Shodan、Censys」ですが、インターネット上に公開されているサーバーのさまざまな情報を収集しており、検索、閲覧が可能なサービスです。Shodan での閲覧イメージを次ページに示します。

これらの情報の収集は、「Shodan、Censys」が独自に行なっています。その情報収集技術の 1 つが、ポートスキャンです。「ポートスキャン」はネットワークに接続されている、通信ポート全てに特定のデータを送信して、その応答状況から調査を実施することです。「1. 基礎知識編」では、「Shodan、Censys」で得られる情報をイメージすることを目的に、「ポートスキャン」を試行します。

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。OA 端末（社内ネットワーク接続機器）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いたします。

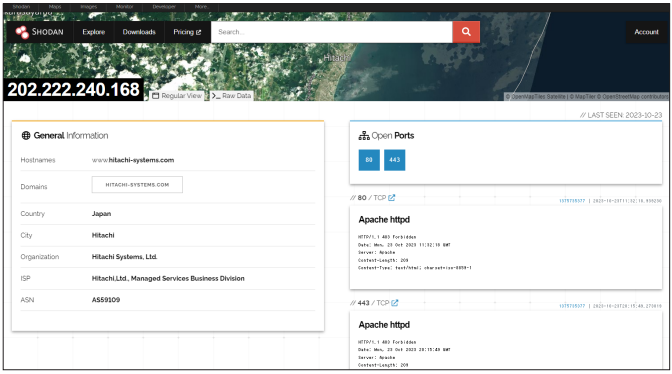


図 Shodan での閲覧イメージ

2. 準　備

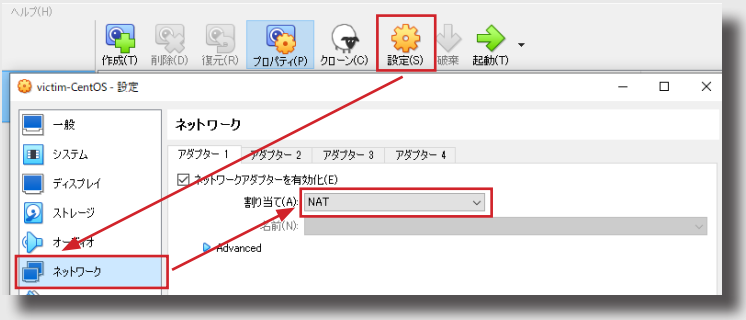
2.1 Nmap を実行する CentOS の準備

Nmap を実行する CentOS を準備します。
CentOS は、「Let's Try HDD 保全！1. 準備編」で作成しています。作成済みの方はそちらをご利用ください（スナップショット機能を活用してください）。
本稿ではじめて作成する方は、下記 URL よりバックナンバーを参照してください。

・Let's Try HDD 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

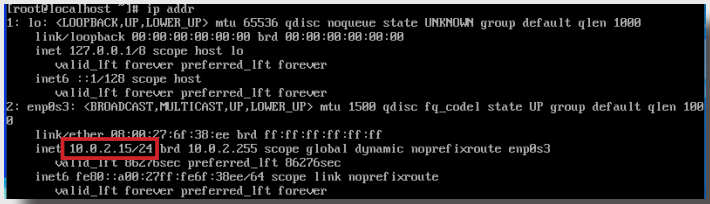
「設定」→「ネットワーク」→「アダプター１」の割り当てが、「NAT」または「ブリッジアダプター」など、インターネットに接続できる設定となっていることを確認します。



また、CentOS 起動、ログイン後に以下のコマンドを入力し、IP アドレスが取得できていることを確認します。

```
# ip addr
```

例では、ネットワークインターフェース enp0s3 に IP アドレス「10.0.2.15/24」が付与されていることが確認できます。

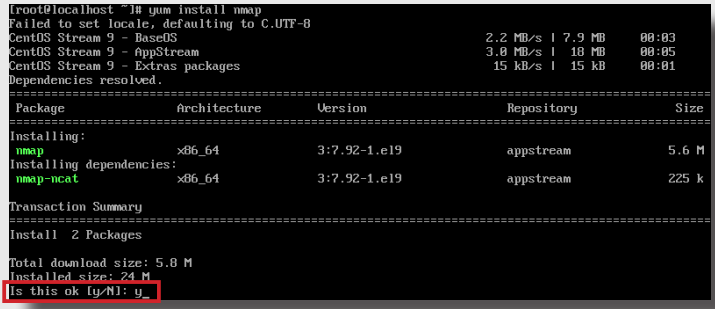


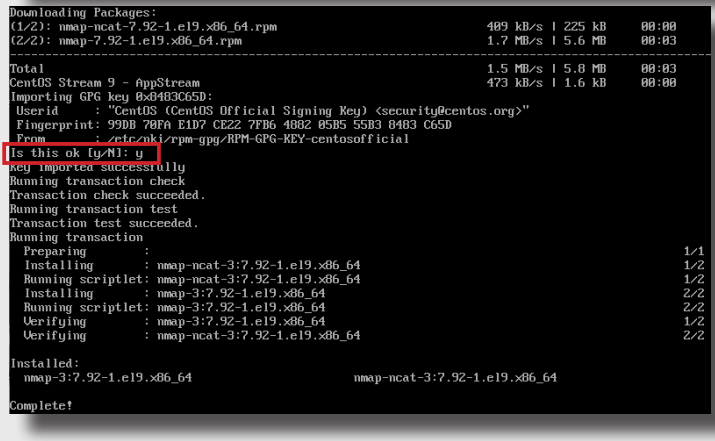
2.3 Nmap のインストール

CentOS に Nmap をインストールします。Nmap は、もっとも著名なセキュリティスキャナーであり、古くから利用されています。主に、コンピューターネットワークにおいて、対象のコンピューターの複数のポートに対して接続要求を行ない、ポートの状況を確認する、ポートスキャンに利用されます。CentOS にログインし、以下のコマンドを実行し、Nmap をインストールしてください。

```
# yum install nmap
```

実行結果は以下のとおりです。インストール中に確認が出ましたら、「y」と入力し、インストールを続行します。図のように「Complete」が表示されれば完了です。





3. ポートスキャンの実施

3.1 TCP スキャンの実施

自身のサーバーに対してポートスキャンを実施します。まずは、TCP ポートに関するポートスキャンを実施します。
以下のコマンドを実行してください。

```
# nmap -sT 127.0.0.1
```

```
[root@localhost ~]# nmap -sT 127.0.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-06 21:15 JST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00071s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.60 seconds
```

「TCP 22 番ポート」が開いていること（接続できること）が確認できました。22 番ポートは、「ssh」と表示されています。次に、以下のコマンドの実行してください。

```
# nmap -sTV 127.0.0.1
```

```
[root@localhost ~]# nmap -sTV 127.0.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-06 21:15 JST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00050s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.7 (protocol 2.0)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

V オプションは、バージョンを表示するオプションです。このオプションにより、22 番ポートで稼働しているサービスは、OpenSSH 8.7 であることがわかりました。

3.2 UDP スキャンの実施

次に、UDP ポートに関するポートスキャンを実施します。
以下のコマンドを実行してください。

```
# nmap -sUV 127.0.0.1
```

```
[root@localhost ~]# nmap -sUV 127.0.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-06 21:17 JST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000097s latency).
All 1000 scanned ports on localhost (127.0.0.1) are in ignored states.
Not shown: 1000 closed udp ports (port-unreach)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.16 seconds
```

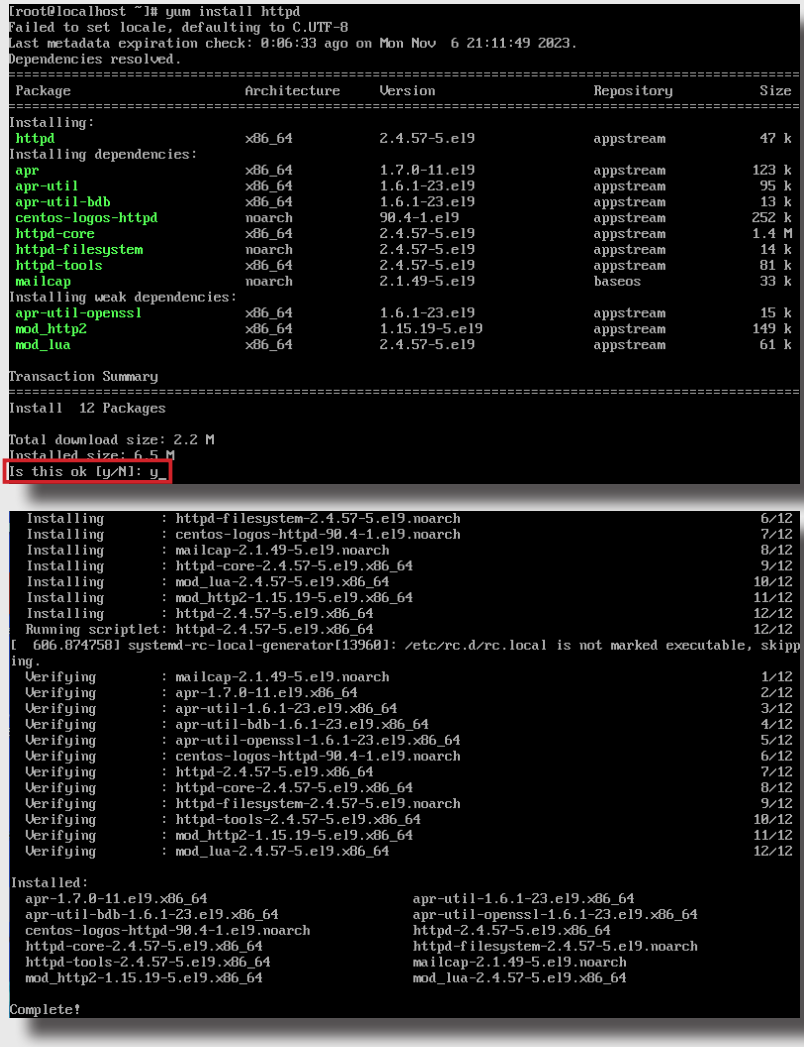
UDP ポートが開いていないこと（接続できないこと）が確認できました。

3.3 httpd（Apache）のインストール

httpd（Apache）をインストールします。
以下のコマンドを実行してください。

```
# yum install httpd
```

インストール中に確認が出ましたら、「y」と入力し、インストールを続行します。図のように「Complete」が表示されれば完了です。



```
[root@localhost ~]# yum install httpd
Failed to set locale, defaulting to C.UTF-8
Last metadata expiration check: 0:06:33 ago on Mon Nov  6 21:11:49 2023.
Dependencies resolved.
================================================================================
 Package                 Architecture    Version                 Repository    Size
================================================================================
Installing:
 httpd                   x86_64          2.4.57-5.el9            appstream     47 k
Installing dependencies:
 apr                     x86_64          1.7.0-11.el9            appstream    123 k
 apr-util                x86_64          1.6.1-23.el9            appstream     95 k
 apr-util-bdb            x86_64          1.6.1-23.el9            appstream     13 k
 centos-logos-httpd      noarch          90.4-1.el9              appstream    252 k
 httpd-core              x86_64          2.4.57-5.el9            appstream    1.4 M
 httpd-filesystem        noarch          2.4.57-5.el9            appstream     14 k
 httpd-tools             x86_64          2.4.57-5.el9            appstream     81 k
 mailcap                 noarch          2.1.49-5.el9            baseos        33 k
Installing weak dependencies:
 apr-util-openssl        x86_64          1.6.1-23.el9            appstream     15 k
 mod_http2               x86_64          1.15.19-5.el9           appstream    149 k
 mod_lua                 x86_64          2.4.57-5.el9            appstream     61 k

Transaction Summary
================================================================================
Install  12 Packages

Total download size: 2.2 M
Installed size: 6.5 M
Is this ok [y/N]: y_
```

```
  Installing       : httpd-filesystem-2.4.57-5.el9.noarch                   6/12
  Installing       : centos-logos-httpd-90.4-1.el9.noarch                   7/12
  Installing       : mailcap-2.1.49-5.el9.noarch                            8/12
  Installing       : httpd-core-2.4.57-5.el9.x86_64                         9/12
  Installing       : mod_lua-2.4.57-5.el9.x86_64                           10/12
  Installing       : mod_http2-1.15.19-5.el9.x86_64                        11/12
  Installing       : httpd-2.4.57-5.el9.x86_64                             12/12
  Running scriptlet: httpd-2.4.57-5.el9.x86_64                             12/12
[  606.874758] systemd-rc-local-generator[13960]: /etc/rc.d/rc.local is not marked executable, skipp
ing.
  Verifying        : mailcap-2.1.49-5.el9.noarch                            1/12
  Verifying        : apr-1.7.0-11.el9.x86_64                                2/12
  Verifying        : apr-util-1.6.1-23.el9.x86_64                           3/12
  Verifying        : apr-util-bdb-1.6.1-23.el9.x86_64                       4/12
  Verifying        : apr-util-openssl-1.6.1-23.el9.x86_64                   5/12
  Verifying        : centos-logos-httpd-90.4-1.el9.noarch                   6/12
  Verifying        : httpd-2.4.57-5.el9.x86_64                              7/12
  Verifying        : httpd-core-2.4.57-5.el9.x86_64                         8/12
  Verifying        : httpd-filesystem-2.4.57-5.el9.noarch                   9/12
  Verifying        : httpd-tools-2.4.57-5.el9.x86_64                       10/12
  Verifying        : mod_http2-1.15.19-5.el9.x86_64                        11/12
  Verifying        : mod_lua-2.4.57-5.el9.x86_64                           12/12

Installed:
  apr-1.7.0-11.el9.x86_64                  apr-util-1.6.1-23.el9.x86_64
  apr-util-bdb-1.6.1-23.el9.x86_64         apr-util-openssl-1.6.1-23.el9.x86_64
  centos-logos-httpd-90.4-1.el9.noarch     httpd-2.4.57-5.el9.x86_64
  httpd-core-2.4.57-5.el9.x86_64           httpd-filesystem-2.4.57-5.el9.noarch
  httpd-tools-2.4.57-5.el9.x86_64          mailcap-2.1.49-5.el9.noarch
  mod_http2-1.15.19-5.el9.x86_64           mod_lua-2.4.57-5.el9.x86_64

Complete!
```

インストールが完了したら、httpd サービスを起動し、起動を確認します。
以下のコマンドを実行してください。

```
# systemctl start httpd.service
```

サービスが起動しているかは以下のコマンドで確認できます。

```
# systemctl status httpd.service
```

active（running）と表示されることを確認してください（コマンドプロンプトに戻る際は、キーボードから「q」を入力します）。

```
[root@localhost ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: active (running) since Mon 2023-11-06 21:23:18 JST; 8s ago
       Docs: man:httpd.service(8)
   Main PID: 14213 (httpd)
     Status: "Started, listening on: port 80"
      Tasks: 213 (limit: 11105)
     Memory: 24.9M
        CPU: 331ms
     CGroup: /system.slice/httpd.service
             ├─14213 /usr/sbin/httpd -DFOREGROUND
             ├─14214 /usr/sbin/httpd -DFOREGROUND
             ├─14215 /usr/sbin/httpd -DFOREGROUND
             ├─14216 /usr/sbin/httpd -DFOREGROUND
             └─14217 /usr/sbin/httpd -DFOREGROUND

Nov 06 21:23:17 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Nov 06 21:23:18 localhost.localdomain httpd[14213]: AH00558: httpd: Could not reliably determine th
Nov 06 21:23:18 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Nov 06 21:23:18 localhost.localdomain httpd[14213]: Server configured, listening on: port 80
```

3.4 TCP スキャンの再実施

httpd（Apache）をインストールし、サービスを起動しました。そのため、http のウェルノウンポートである TCP 80 番ポートまたは、https のウェルノウンポートである TCP 443 番ポートが開いているはずです。以下のコマンドを入力し、再度 TCP スキャンを実行してください。

```
# nmap -sTV 127.0.0.1
```

```
[root@localhost ~]# nmap -sTV 127.0.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-06 21:25 JST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00096s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.7 (protocol 2.0)
80/tcp open  http    Apache httpd 2.4.57 ((CentOS Stream))

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.60 seconds
```

サービス Apache httpd2.4.57 が、TCP 80 番ポートを開いていることが確認できました。

3.5 httpd の設定変更

次に、httpd の設定を変更します。
次ページのコマンドを実行してください。

```
# cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.backup
# systemctl stop httpd.service
# echo ServerTokens ProductOnly >> /etc/httpd/conf/httpd.conf
# echo ServerSignature off >> /etc/httpd/conf/httpd.conf
# systemctl start httpd.service
```

これらの設定で、サーバーのバージョン情報の表示を抑制します。
サービスが起動しているかどうかは以下のコマンドで確認できます。

```
# systemctl status httpd.service
```

3.6 TCP スキャン（バージョン表記あり）の再実施

以下のコマンドで、再度 TCP スキャンを実施してください。

```
# nmap -sTV 127.0.0.1
```

```
[root@localhost ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: active (running) since Mon 2023-11-06 21:23:18 JST; 8s ago
       Docs: man:httpd.service(8)
   Main PID: 14213 (httpd)
     Status: "Started, listening on: port 80"
      Tasks: 213 (limit: 11105)
     Memory: 24.9M
        CPU: 331ms
     CGroup: /system.slice/httpd.service
             ├─14213 /usr/sbin/httpd -DFOREGROUND
             ├─14214 /usr/sbin/httpd -DFOREGROUND
             ├─14215 /usr/sbin/httpd -DFOREGROUND
             ├─14216 /usr/sbin/httpd -DFOREGROUND
             └─14217 /usr/sbin/httpd -DFOREGROUND

Nov 06 21:23:17 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Nov 06 21:23:18 localhost.localdomain httpd[14213]: AH00558: httpd: Could not reliably determine th
Nov 06 21:23:18 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Nov 06 21:23:18 localhost.localdomain httpd[14213]: Server configured, listening on: port 80
```

実行の結果、Apache httpd が 80 番ポートを開いていることは確認できますが、Apache httpd のバージョン番号まではわかりません。Nmap が、Apache httpd の HTTP レスポンスに含まれるバージョン番号をもとに、Apache httpd のバージョンを確認していたことが確認できました。

4. おわりに

今回はここまでとなります。

ポートスキャンは、IoT 検索エンジンの情報収集技術の 1 つです。IoT 検索エンジンは、存在するすべての IP アドレスに対して順に、ポートスキャンなどのスキャン行為を実行し、情報を収集しています。今回は、「基礎知識編」として、セキュリティスキャナー Nmap を用いてポートスキャンを試行しました。

次回は「所有サーバー確認編」として、自身が管理している IP アドレス等がわかるサーバーが「Shodan、Censys」といった IoT 検索エンジンでどのように見えるのかを確認します。

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.56

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