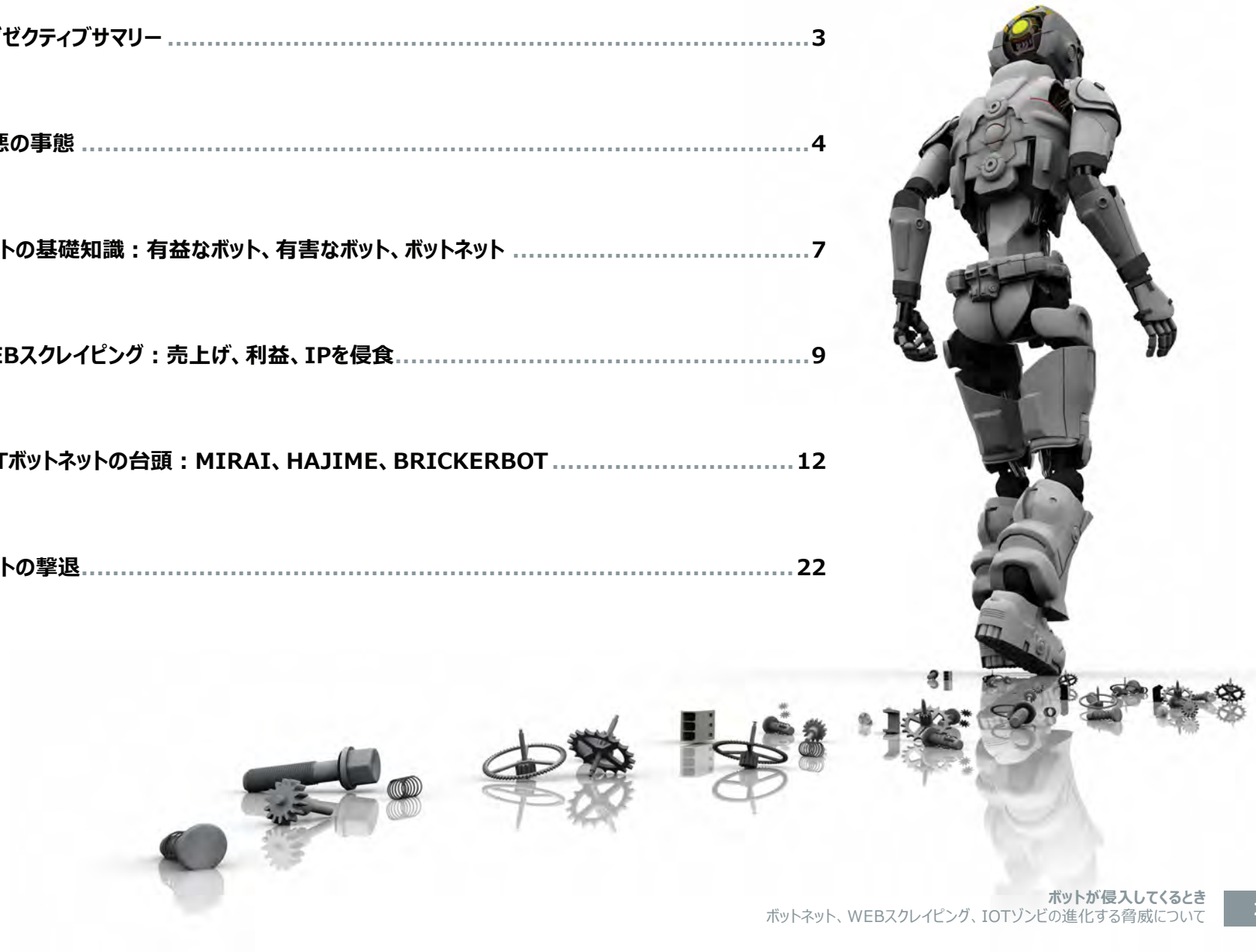
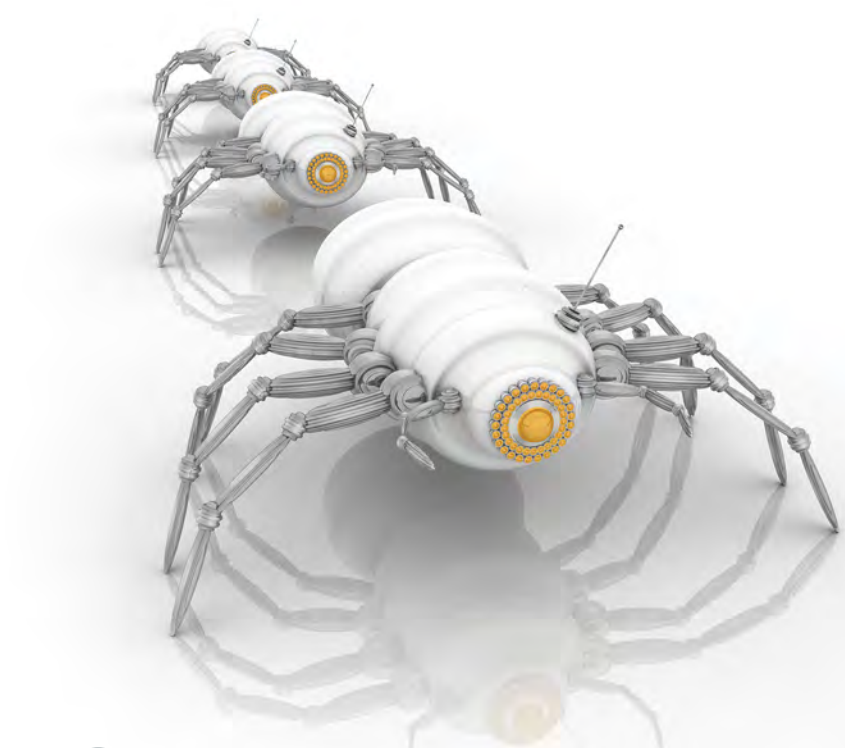
radware

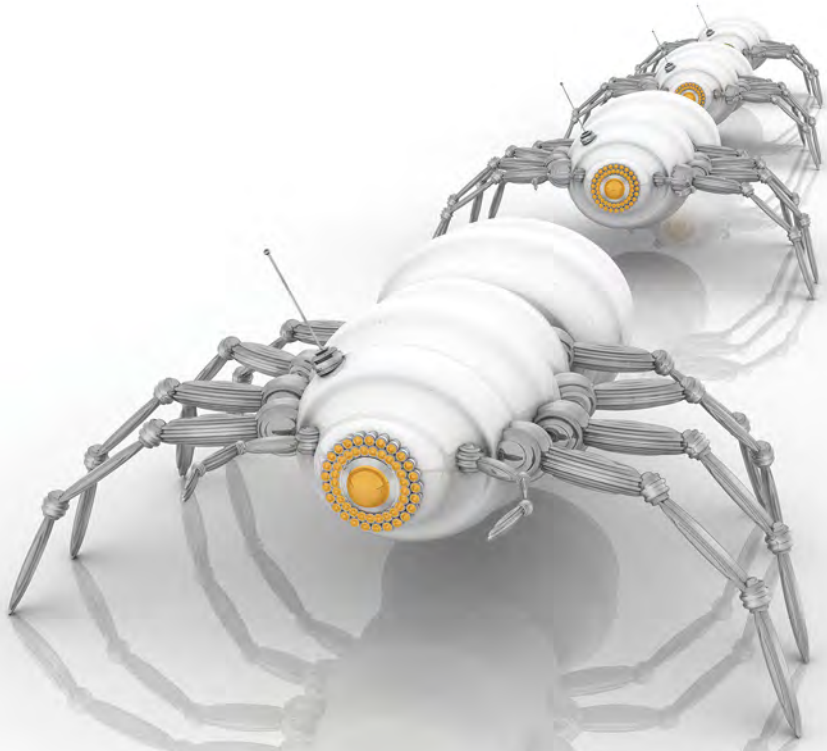
ボットが侵入してくるとき

ボットネット、Webスクレイピング、IoTゾンビの進化する脅威について

目次







エグゼクティブサマリー

このe-bookに記載された情報は、皆さんが読んでいただくときには時代遅れになっているかもしれません。つまり、それほどボット、ボットネット、Webスクレイピング、そしてIoTを悪用した脅威は素早く、そして劇的に進化しているということです。

では、これを読むのは時間の無駄になるのでしょうか？いえ、そんなことはありません。

たちの悪いボット、つまりIoT（Internet of Things）を攻撃の武器として悪用するものなどは、最も速く拡大し変化している脅威の1つです。2016年に発生した「Mirai」による攻撃では、ハッカーがボット部隊を展開して安全性に欠けるIoTデバイスを制御してしまうと、何が起こってしまうか明らかになりました。ただし、そのような攻撃はもはや最悪のシナリオではないかもしれません。ラドウェアでは、最近新たなタイプのボットを発見しました。それは「BrickerBot」で、永続的なサービス拒否（PDoS：Permanent Denial-of-Service）攻撃を行って、既に感染しているデバイスを「守ろう」とするものです。その通り、感染したIoTデバイスが別のIoTボットネットに組み込まれないように、BrickerBotは完全にそれらをシャットダウンします。狙われたデバイスは実用性も価値もない「レンガ（操作不能）」以外の何物でもなくなります。

間違いを犯さない：これは脅威として現在進行している物語であり、検知と緩和ソリューションは絶え間なく変化します。次のような現状を把握し、知識と防御態勢を整えてください。

- これらの脅威を実現するに至ったテクノロジーのトレンドと勢力
- Webスクレイピングとそれが売上げ、利益、IPを浸食する方法
- ボットの基礎知識（すべてが有害なわけではありません）
- これまで判明している世界で最も特筆すべき3つのボットについて

最悪の事態

2016年は、「Mirai」として知られるIoTボットネットによって、Krebs、OVH、そしてDynが攻撃を受けました。Miraiは初のIoTボットネットでも、最も洗練されたIoTボットネットでもありませんでしたが、非常に効果的に標的を陥れました。このような攻撃は、IoTボットネットとDDoSの歴史において1つの節目となり、ネットワーク、システム、データの保護に責任を持つ者すべてに警鐘を鳴らしました。

Miraiは、IoTとDDoSの脅威の状勢において変曲点を1つ示しました。ただし、Miraiを「成功」に導いたのは、セキュリティ対策されずに接続された多数のデバイス、法人や個人でのクラウドへの依存、多数の廉価なツールに簡単にアクセスしてDDoSやその他サイバー攻撃を始められる活発なハッカーエコノミー、等その他多くの傾向や情勢によるものです。

リスクが大きく、さらに危険の度合いが高まるトレンドを示す「最悪の事態」について詳しく見ていきましょう。

IOTデバイス: 急速な拡大・・・

「モノのインターネット（Internet of Things）」という用語は、1999年に生まれました。部品コストが下がり、法人・個人の要求が加速度的に伸びた2014～2015年に初めて、IoT自身が変曲点に達しました。電球や洗濯機から医療機器に至るまで、多種多様の「モノ」がインターネットに接続され、マシン-to-マシンソリューションが主流となっています。[1]

2016年には、インターネット接続されたモノの数は、インターネットに接続したユーザーのほぼ倍になり、接続デバイス数はインターネット利用者数よりはるかに速く拡大しています。ある情報によると、導入されるIoT数は2020年には200億に達すると言われています。

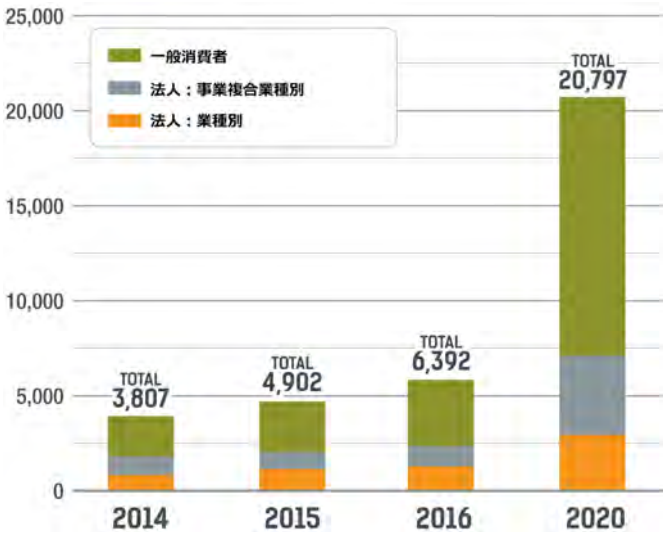


図1: カテゴリー別導入済みIoTユニット数（単位：100万）
出典: http://www.gartner.com/newsroom/id/3165317

IoT機器のセキュリティ上の課題

なぜ、IoTデバイスがサイバー攻撃の格好の標的となるのでしょうか？主に、次の4つの要因が考えられます。

1. **必要最小限のOS** このようなデバイスは、Linuxオペレーティングシステム上で動作することがありますが、組み込み型または必要最小限の機能に絞ったバージョンが使用されており、マルウェアによる不正アクセスが比較的に容易です。
2. **制約のないアクセス** 「モノ」でインターネットアクセスが可能な場合、通常そのアクセスは、回線利用でのフィルタリングや制限の制約を受けません。
3. **基本的なセキュリティの欠如** 必要最小限のOSと処理能力のみを備えたこれらのデバイスでは、監査のような標準的なセキュリティ機能を十分には持ち合わせていません。その結果、デバイスオーナーは、大部分の不正アクセスに気づくことさえありません。
4. **再利用コンポーネント** デバイスメーカーは、さまざまなデバイスのハードウェアやソフトウェアの一部を再利用する場合があります。開発時間の短縮を目的にはしていますが、この手法が、デフォルトのパスワードや脆弱性がデバイスクラス間のみならず、メーカー間でも共有される結果も招いてしまいます。

クラウドへの移行とサーバーレスコンピューティングの出現

より多くの企業がクラウドに移行するため、より多くのコアアプリケーションがパブリッククラウドにホストされるようになり、攻撃者が利用できる標的の数もますます増加しています。世界は仮想マシン（Infrastructure-as-a-Service)からアプリケーション(Software-as-a-Service)に移行しましたが、次の大ブームはサーバーレスコンピューティング(Function-as-a-Service)になりそうです。

サーバーレスコンピューティング（FaaS）は、APIエコノミーではまさに自然な流れで、既にその時代はやってきています。ハイパースケール・クラウドアプリケーションが、マイクロサービスアーキテクチャを利用するようになったため、APIエコノミーは一般に利用可能なAPIとして、これら内部マイクロサービスを具体化します。

1 出典: http://radar.oreilly.com/2014/06/the-internet-of-things-inflection-point.html

サーバーレスコンピューティングが始まると、より多くのクラウドアプリケーションは、実質的に多くのAPIに依存するようになります。そして、次は複雑に相互接続した機能の世界が生まれますが、これは二次的損害を与えることが可能であったDyn攻撃さえ超越する相互依存レベルです。

つまり、クラウドアプリケーションに対するパブリックアクセスのシングルポイントを、機能のモジュラーセットに変えると、標的の数が増えることになります。さらに、不正アクセスされた機能やサービスの1件の被害により、影響を受けるサービスや企業は数件から多数にまで増加します。これで攻撃者が攻撃する標的はさらに増え、攻撃が成功した場合の報酬が増えます。

ハッカーエコノミーの熟成

今日では、誰もが、極めて限られた技術ノウハウしか持たない人でさえ、オンラインマーケットでツールを購入して攻撃を実行できるまでになっています。暗号通貨では追跡不可能なデジタル決済が可能で、一方で従来からの経済はこれらマーケットの成長を推進しています。サービスに対する要求は、今や供給を上回っており、DDoS-as-a-Serviceプロバイダーは年間10万ドル以上も利益を上げられます。

攻撃（手段）の購入は、驚くほど安価です。クリアネット（Clearnet）では、月額わずか$19.99で、DNS、SNMP、SYN、スローGET/POSTアプリケーション層DoS攻撃など、多数の攻撃ベクトルを利用して30日間に20分のバースト攻撃を実行します。攻撃者が行うことは、アカウントの作成、プランの選択、ビットコイン（Bitcoin）での支払い、そして攻撃ハブにアクセスしてポート、時間、方法別に標的を狙うことです。より高度で大規模なボットネットは、ダークネット（Darknet）でも販売されています。[2]

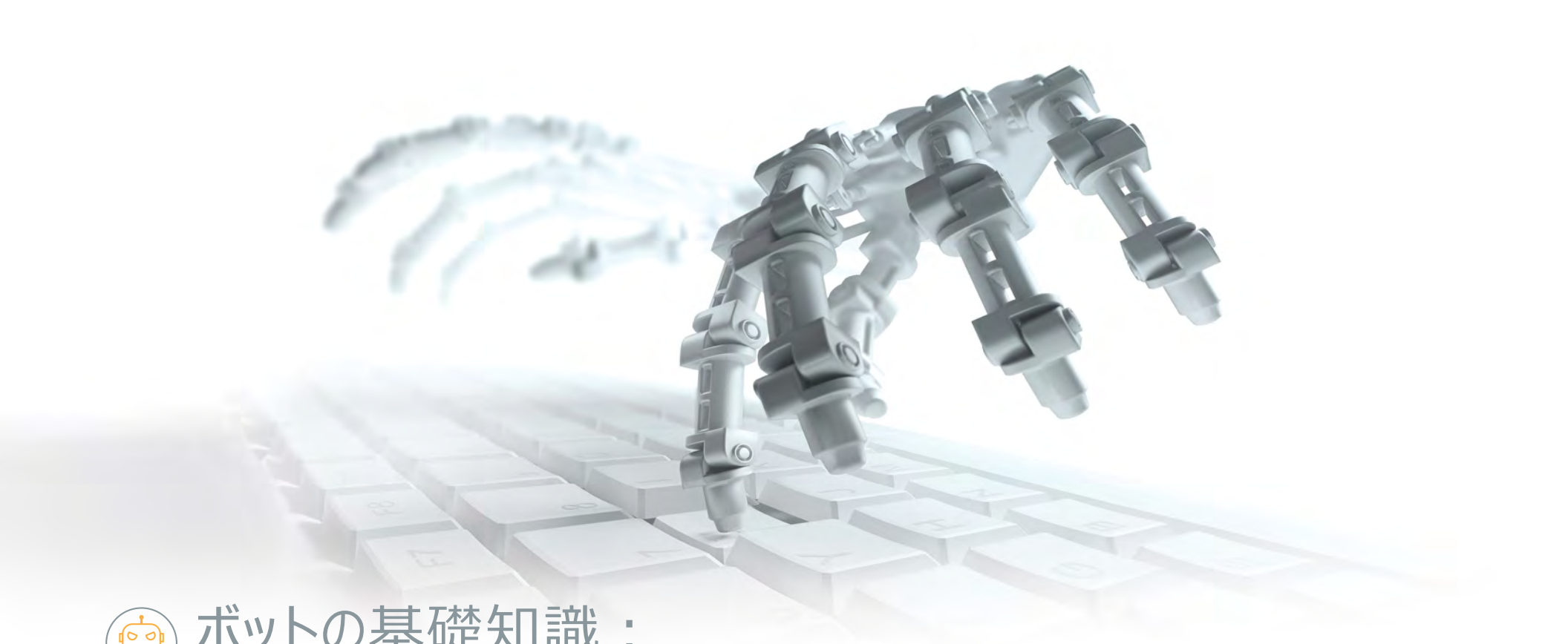
図2: 購入可能な攻撃の例

なぜこれらの攻撃に対して支払ってしまうのか？ ラドウェアでは、3つの主な要因を特定しました。

- **利益** ランサム（身代金要求）活動は、手っ取り早くお金を奪い取る方法です。
- **注意をそらす** マルチベクトル攻撃では、DDoS攻撃で有効となる偽装行為を行い、その先の標的型攻撃かつ、またはデータ抽出を隠します。
- **妨害と改変** この攻撃は、意見が合わない組織に対して罰を与えようとするハクティビストか、または競合他社を陥れようとする組織が実行します。

サイバーランサム攻撃、およびハッキングエコノミーについての詳細は、ラドウェア発行の「サイバー身代金要求対策ガイド 拡大を続けるランサムウェアとRDoSの脅威 – その対策」をご覧ください。

2 攻撃（手段）取引市場についての詳細は、ラドウェアのブログをご覧ください。



ボットの基礎知識：有益なボット、有害なボット、ボットネット

「ボット」とは？

一般的には、「ボット」は「ロボット」の省略形です。インターネットの流れにおいては、完全形態は「Webロボット」または「インターネットロボット」で、一連の自動タスクを実行するようプログラムされたコンピュータシステムを指します。また、ボットは「ゾンビ」と呼ばれる場合もあります。

図3にあるように、Webサイトやサービス上で情報を収集するといった好意的で有益なタスクを実行するボットもある一方で、コンピュータデバイスの乗っ取りや感染を目的として脅威の行為者が作成するボットもあります。ボット攻撃の標的は、個人のコンピュータ、スマートフォン、タブレットからサーバーや接続された「モノ」までに至ります。

ボットはオンラインの世界では巨大なプレーヤーに [3]

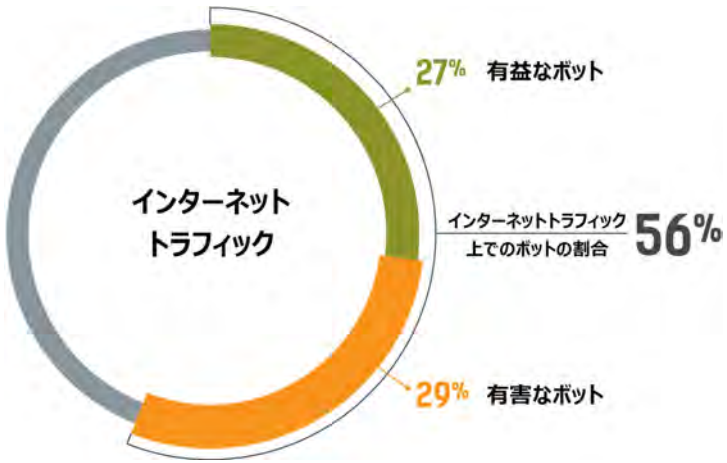
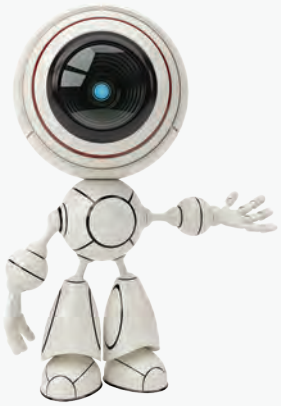


図3: オンラインの世界でのボット

3 https://areyouahuman.com/downloads/GoodBotsvBadBots_FINAL.pdf



有益なボット

インターネットをよりよいものにするために役立つ

有益なボットがサポートするもの

- Webインデックス
- 旅行業界情報まとめサイト
- ニュースとりまとめ
- 自動商取引
- 有害なボットからの防御

例:

- スパイダーボット
- メディア/RSS/データボット
- Vigilante™ ボット
- トレーディングボット
- チャッターボット

有害なボット

インターネットで個人および法人のリスクを生み出す

有害なボット:

- DDoS攻撃を通じて損害を生じさせる
- WebクローリングやWebスクレイピングでデータを盗用
- アプリケーション、サーバー、ネットワークをスキャンして悪用
- スパムやフィッシング活動の実行
- アドクリック詐欺の実行

例:

- クリッカーボット
- ハッカーボット
- ダウンロードボット
- ビットコインマイニングボット
- スパムボット
- マルウェア/ウィルスボット
- スクレイパーボット

図4: 有益なボットと有害なボットの例

ボットネットとは？

ボットネットは、指揮統制（Command and Control:「CnC」と省略されるが、「C&C」や「C2」とも略される）サーバーに監視される一群のボットです。各ボットネットは、同じボットネット内のすべてのボットを組織化して制御する1つのCnC（有効な場合はさらに多くの）があります。ボットネット内の個々のボットは、CnCに「Call Home（攻撃者との通信）」を行うようプログラムされており、そこで指示や命令が送られます。ボットネットは究極の攻撃ツールです。その理由は次のとおりです。

- ボットは攻撃者と直接関係がありません。
- ボットはCnCサーバー経由で自動化されます。
- ボットは地理的に分散しています。
- ボットは使い捨てで、必要に応じて簡単に置き換えられます。
- ボットは柔軟性が高く、広範囲の不正な活動に使用できます。

図5にあるように、ボットネットのサイズは大きくなっており（ボットネットが大きくなり、同時により高性能なデバイスに影響をおよぼすようになり）、結果的にさらに高度な攻撃となっています。

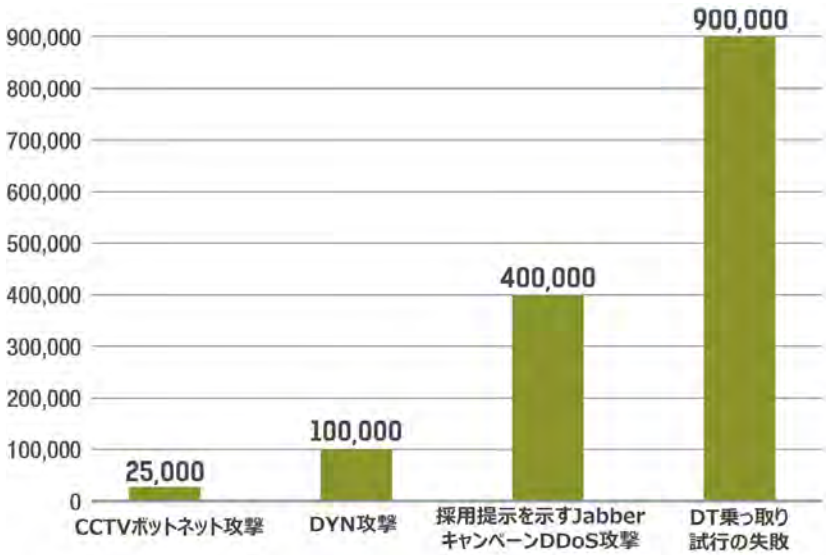


図5. IoTボットネットの推定サイズ 2016年
ラドウェアの緊急対応チームの調査による

WEBスクレイピング:
売上げ、利益、IPを侵食

Webスクレイピングは、Webサイトからデータを収集し、多岐にわたる目的で使用される一種のソフトウェアを指します。Webスクレイピングには、主に次の5つの用途があります。

- コンテンツスクレイピング(正規のWebサイトからオリジナルコンテンツを取り出し、所有者に知らせず、また許可を得ずに別のサイトに掲載します)
- 調査
- 価格比較
- データモニタリング(天気、株式など)
- Webサイトの変更検知

コンテンツスクレイピングは、知的財産かつ、またはデータの盗用など組織にとっては多大なリスクをもたらします。さらに、コンテンツスクレイピングを実行するボットは、サービス拒否（DoS：Denial-of-Service）状態を引き起こす連続的なリクエストを多数実行する場合があります。また、その被害に遭った企業は、この情報収集行為や価格比較サイトの影響、または情報漏洩が理由となり利益を失う可能性があります。

ブラウザで表示できるものはスクレイピング可能

オンライン資産のスクレイピング（データ盗用）を防止するには何をするべきでしょうか？機密と見なされる情報には、公開して初めて有益となるものがあります。一般的な例として、航空運賃、ホテルの宿泊費、医師のリストなどです。場合によっては、サイトでデータを分かりにくくしようとします。実際のデータを、ダイナミックグリッド、AJAXかつ、またはWebSocketを使用してダウンロードする場合は、データレコードのスクレイピングを大幅に困難にさせようとします。

ブラウザがスクリーンに表示できるものはすべて、体系化されたドキュメント・オブジェクト・モデル（DOM：Document Object Model）の一部としてメモリー内に保持され、そのコンテンツはスクリプトライブラリやプログラミングライブラリからアクセスできます。

皮肉なことに、一般に使用されるスクレイピングツールの大半は、まったく別の目的（品質保証）で設計されています。Seleniumとそれに類似したツールは、Webアプリケーションのテストに使用されます。それらのツールは、ユーザーの相互作用のシミュレートおよび自動化が可能で、Webアプリケーションからのテスト結果を受け取ります。ただし、Seleniumとそれに類似したツールのその同じ機能を使用すると、一般に利用できるあらゆるデータのスクレイピングを自動化できてしまいます。また、ヘッドレスクライアント、またはリアルブラウザクライアントは、ボット検知をさらに困難にさせるために使用できます。これらの技術は、ユーザーの振る舞いの模倣、問題箇所の回避、他のボット検知アルゴリズムの妨害に利用されます。

スクレイピングサービス：GOOGLE検索できてしまう

分散型サービス拒否攻撃（DDoS-as-a-Service）ツールの取得が容易になったように、Webスクレイピングのオンラインサービスを利用することも手軽に、簡単になりました。

ボット攻撃を受けて：最前線に聞くスクレイピング事例

オンラインのバーゲン情報を入手することは、Webスクレイピングでは最も一般的な利用目的の1つです。Webスクレイピングツールでは、オンライン価格の追跡や、いったん価格が下がった時を特定して、多数のリクエストを作成することは比較的容易にできます。人と比べて、ボットはリクエストの生成、分あたりのリクエストの大量作成（実際のもの、偽装のものを問わず）ははるかに効率的に行えます。発生し得る結果：オンラインストアの在庫が空になるため、不正流通業者はより高い価格で商品を再販できます。[4]

おそらく、皆さんこのような経験をしています。もうすぐ開催されるコンサートについてかつて耳にしたことを考えてみましょう。チケットがオンラインで購入可能となったまさにその瞬間、あなたはチケットを購入しようとします。ですが、気に入った席は既に販売済み！後に、チケット仲介Webサイトで、その席が5～8倍の値段で売られているのを見つけます。それがWebスクレイピングでもたらされた状況です。

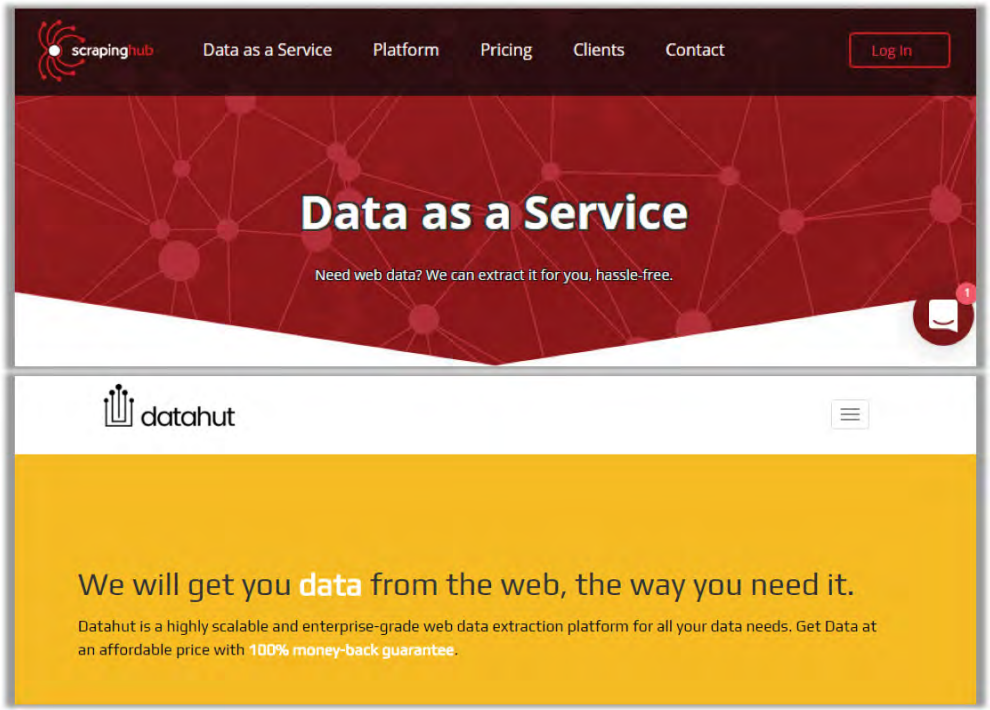


図6：Webスクレイピングサービスの例

4不正流通業者は、メーカー公認取引チャネル外で商品を売買します。

他の事例…

航空会社も、一般にWebスクレイピングの標的となります。ラドウェアの顧客である米国拠点の大手航空会社では、この種のサイバー攻撃が驚くべき頻度で発生しました。ボットは、特定のフライト、航路、チケットのクラスを「スクレイピング」するようプログラムされていました。偽装購入者として振る舞うボットは、継続的に該当するチケットの予約を行いながらも決して予約処理を完了せずにいたため、航空会社は実際の顧客に座席チケットを販売できませんでした。
実質的には、この航空会社の座席チケットの在庫が人質に取られ、ますます多数の便が（売られるはずであった）席を空にして飛び立ったのでした。[5]

英国では、有名ブランドのWebサイトが課金制で運営されていました。同社は、自社のWebサイト全体がスクレイピングされ、中国のホスティングサイトに無料で売りに出されるまでWebスクレイピングを気にかけていませんでした。

何年も前に、ラドウェアはショッピングカートに$99,000の商品を入れて、レジ段階に進んでいる競合企業に悩まされていたオンラインストアを援助したことがあります。予期できない障害だったのでしょうか？実は、たとえ競合企業が実際にレジ処理を終えていなくても、在庫は減ったように見えていました。実際の顧客にとっては、すべてが「在庫なし」と表示されており、入荷待ちをする必要がありました。さらに、このオンラインストアは競合企業が価格比較のためにサイトにアクセスしていたことを突き止めました。ラドウェアのWebパフォーマンス最適化ソリューションでスクレイパーボットをブロックした後は、サイトのWebトラフィックの帯域幅は66%減少しました。これではっきりしたことは、どれほどの帯域幅が有害なボットに浪費されていたかということです！そのようなボットが排除されると、Webサイトはスピードとパフォーマンスが2倍に改善されました。

図7: 他のWebスクレイピングサービスの例

5 出典: ラドウェアのグローバルアプリケーション&ネットワークセキュリティレポート 2015-2016

IOTボットネットの台頭：MIRAI, HAJIME, BRICKERBOT

これまでに述べたとおり、ボットはボットネットという名称で知られる集団に取り込まれ、IoT（Internet of Things）は、驚くほど低いレベルのセキュリティを備えた接続デバイスに囲まれています。この2つが組み合わさると、つまり数万や数十万のIoTデバイスが同じボットに感染し、大規模かつ分散したIoTゾンビ軍団に変化すると、何が起こるか想像するのは難しくありません。

これは、ラドウェアが以前から予測していたことであり、2016年10月にMiraiボットネットによって明確に現実のものとなりました。不吉なことに、日本語の「未来」にちなんで名付けられたMiraiは、たとえ単純で洗練されていないボットであっても、どれほどの甚大な損害が与えられるかを示すことになりました。**Miraiをブルートフォース（総当たり）攻撃ボットとしてみなします。大規模で愚か、かつ危険です。**

それから間もなく、別のIoTボットネットが発生しました。Hajimeと呼ばれるこのボットネットでは、Miraiで使用されているいくつかの技術がさらに洗練されています。攻撃を仕掛けるためにボットの集団を捕らえると言うより、HajimeはIoTデバイスで、権限をさらに主張するように設計されているようです。今までのところ、Hajimeは既存のボットを追い払い、ポートを閉じ、デバイスに身を潜めています。その最終目的は未だに不明ですが、世界的な被害の可能性は迫っています。**Hajimeを隠密作戦用ボットとみなします。ずる賢い上に謎が多く、甚大な被害が発生する恐れがあります。**

ごく最近、ラドウェアの社員Pascal Geenensは、IoTボットネットのまったくの新種を発見しました。BrickerBotと呼ばれることになったこのボットは、これまでとは違うまったく新しい目的を持っています。Miraiは素早く攻撃集団をつなぎ、Hajimeが静かに構築に専念していると思われる一方で、自身の集団では行動を起こさないBrickerBotは、「崇高な」目的を持っています。Janit0rとして知られるその作者は、永続的なサービス拒否（PDoS：Permanent Denial-of-Service）を利用して、セキュアではないIoTデバイスを保護していると主張しています。BrickerBotは単に他のボットを追い出したり、デバイスを乗っ取ったりするのではなく、むしろそれらを「レンガで囲み」ます。そうすることで、デバイスがIoTゾンビ集団の一部として取り込まれるリスクをなくします。もちろん、それらがペーパーウェイトとして以外なんら機能を果たさないことにもなります。**BrickerBotを自警団ボットとしてみなします。感染したIoTデバイスをレンガで囲み、IoTセキュリティが必要であると明らかに警告を発しています。**

これらのボットネットを分析し、その損害の負わせ方について見てみましょう。それらが進化し、新規参入者との競争に勝つことは間違いないと思いますが、これら3つのボットを知ることで、インターネット上で繰り広げられているIoT市場シェア争いを深く理解できるようになり、今後のDDoS攻撃の傾向やIoTベースの脅威をより把握するのに役立つでしょう 。

MIRAI: 大規模かつシンプル。あり得ないほど破壊的

IoTのセキュリティと破滅的なIoTゾンビの脅威に対する不安は、永らく取りざたされてきました。ただし、Miraiは最終的に1つの変曲点となったもので、数十万の感染したIoTデバイスが大規模なDDoS攻撃を起こし得ることを示しました。以降では、2016年に広がったその展開プロセスについて説明しています。

- **9月20日** - 午後8時頃、KrebsOnSecurity.comがサイトのオフラインを目的とした620Gbpsという記録破りの大容量DDoS攻撃の標的となりました。攻撃トラフィックの大部分がGREペイロードで構成されており、これは大規模な容量型攻撃では極めて異例なことでした。
- **9月21日** – 同一タイプのボットネットが、フランスのWebホスティング会社OVH3社を標的とした大規模な容量型攻撃に使用されました。この攻撃の特徴は、攻撃の実行に膨大な数のデバイスが使用された点です。攻撃の標的はOVH社ではなく、同社がホスティングしているゲーミングサーバーのMinecraftTMでした。
- **9月30日** - MiraiボットネットのソースコードはAnna-Sempaiというオンライン名を使用した人物によってHackForums.netで公開され、同年の「看板」ツール・オブ・ザ・イヤーになりました。
- **10月21日** - 多くのFortune 500企業から信頼を得ている米国拠点のDNSプロバイダーDyn社が、「水責め」攻撃として一般に知られている同様のボットネットの攻撃を受けました。この攻撃で多くのサービスが利用不能となり、大規模な接続問題（大半が米国の東海岸周辺）が発生しました。

シンプルながら破滅的

Windows関連のものと比較すると洗練はされていませんが、Miraiはシンプルなボットネットながら、標的を停止させるには効率的かつ効果的であることを証明しました。実際に、オリジナルのMiraiは、高度な感染ベクトルを一切使用していませんでした。その一方で、61のユーザー名/パスワードを組み合わせた限定ディクショナリを使用してTelnetにブルートフォース（総当たり）攻撃を行いました。ポート23でのCnC通信には、シンプルなクリアテキストのTCPベースプロトコルを採用していました。そして、CnC通信が検知されて容易にブラックリスト化されないように、ドメインやドメイン生成アルゴリズム（DGA：Domain Generation Algorithm）を省いていました。さらに、「アップグレード」機能が付いていなかったため、IoTボットは作業の実行に手が込んだ機能を要していないことが分かりました。事実、MiraiのようなIoTボットネットは、使い捨てだと思われます。古いボットネットが不正アクセスされると、それはすぐに捨て去られ、新しいものを簡単に獲得します。

1Tbps以上のトラフィック量を容易に生成できる機能を備えるほかに、Miraiには10種類の攻撃ベクトルがあらかじめ設定されており、そのいくつかは、GREフラッド攻撃を用いてサービスプロバイダーのインフラを攻撃するのに効果があることが実証されています。10種類の攻撃ベクトルの中には、TCP STOMPやDNSリゾルバフラッド（「水責め」)攻撃など、極めて高度な攻撃ベクトルが含まれています。
また、MiraiのDDoS攻撃によって、GREトラフィックやDNS再帰的クエリの正当性の可視化に関して、企業が直面する課題も浮き彫りとなりました。

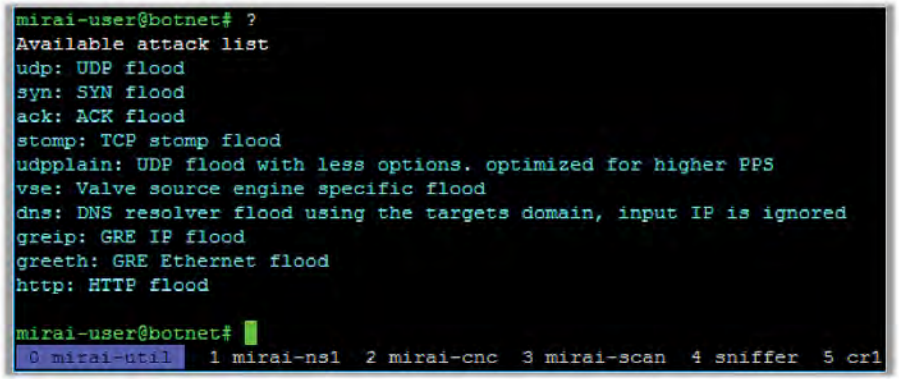


図8: Miraiボットネットのコマンドセット

Miraiはシンプルで、それほど洗練されていないかもしれませんが、ルールを書き換え、IoTへのDDoSボットネット攻撃に新たなリスクがあることをはっきり示すことには成功しました。
IoTの初のオープンソースボットネットとして、Miraiはリアルタイム軽減に関わる体制の在り方に影響をおよぼし、セキュリティは自動化が必須となりました。それは、IoTボットネットが、記録的な大容量攻撃を実行しつつ、あらゆる保護対策を回避するために、継続的に適応する高度なL7攻撃が開始できるからという理由だけではありません。Miraiがオープンソースであるという事実は、つまりハッカーがそれを変更、カスタマイズ、そして改良できる可能性があると言うことであり、インテリジェントな自動化を通してのみ検出可能な、膨大な種類の新たな攻撃ツールを生むことになるのです。

HAJIME: スマートかつ高度な技術。分かりにくい動機

2016年10月16日、Rapidity Networks社のSam Edwards氏とIoannis Profetis氏が、自らが発見し、「Hajime」と名付けた新たなマルウェアについてレポートを発表しました。[6] (Miraiは、日本語で「未来」を意味しており、Hajimeは「始め」を意味しています[7]) このマルウェアの発見は、Miraiのソースコードの公表と、MiraiのKrebsおよびOVH社への攻撃の後です。そして、この新たなマルウェアが世間を賑わす前に、Dyn社やAmazon、CNN、Netflix、Spotify、Twitterなど「有名」顧客リストにあがる各企業を陥れた10月21日の攻撃の出所はMiraiであると考えられました。

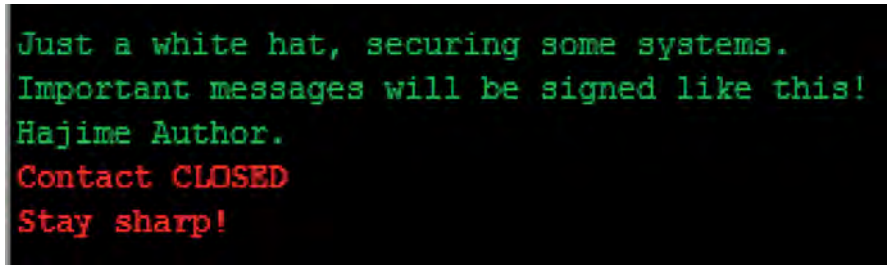


図9: 端末に定期的にメッセージを表示するHajime

この新たなマルウェアは、注目を浴びることなく巧妙に逃れたかのようでしたが、そうではありませんでした。
実際、確実かつ静かに成長を続けています。

CnCチャネルの暗号コードに脆弱性があると伝えたRapidity Networks社によるレポートの後、マルウェアの作者は新たな改良バージョンのボットネットに更新しました。もはや脆弱性のなくなったこのバージョンでは、「Hajime」の名称を採用し、端末に定期的に表示されるメッセージには「Hajime Author（Hajimeの作者）」と表記するようになりました。

6 https://security.rapiditynetworks.com/publications/2016-10-16/hajime.pdf

7 https://www.extremetech.com/internet/248087-meet-hajime-iot-botnet-built-vaccinate-devices-mirai

特に、Hajimeはその発見以降、めざましくIoT市場シェアを伸ばしてきました。ラドウェアのハニーポット（調査目的でハッカーを誘い出して攻撃を探る仕掛け）では、Hajimeによる感染試行数は、全IoTボット活動のほぼ半数を占めています。ラドウェアでは、感染においてHajimeは他の感染したノードを利用して自分自身のマルウェアをダウンロードする場合があることを発見しました。そのため影響範囲は大きくなり、ハニーポットで特定できたHajimeに不正アクセスされたデバイス総数は19,000近くにのぼりました。図10を参照。Kaspersky社によると、ピアツーピア・ネットワークは2017年4月25日までに約300,000デバイスに到達していました。[8]

次世代型IOTボット

Miraiとは異なり、Hajimeは高度で柔軟性が高く、思慮深く設計され、将来においても効果を発揮します。同時に、HajimeはMiraiの感染手法と総当たり攻撃エクスプロイトを利用しています。自分自身で更新（アップデート）する機能を備えるHajimeは、迅速かつ効率的に「豊富な」機能を搭載したメンバーボットを拡大させることができます。指揮制御や更新に使用する分散型ボットネットワークは、トラッカーレスのトレントを使用します。これらのトレントチャネルは、CnCアクティビティをうまく隠すために、毎日変更するいくつかの動的info_hashes値を用いた有名なBitTorrentピアツーピア・ネットワークの最上位に位置しています。BitTorrentを通じた通信はすべて、RC4や秘密/公開鍵を使用して署名および暗号化されます。

図10. Hajimeの地理的分布（各ドットは、ラドウェアが感染デバイスとして特定できた固有のIP）

Hajimeは、拡張モジュールをサポートするモジュラーマルウェアです。
現状の拡張モジュールは、新たな標的の検知と感染のためのスキャン/ローダーサービスを備えています。効率的なSYNスキャナの導入においては、開いているTCP/23(Telnet)ポートやTCP/5358(WSDAPI)ポートを使用して、新たな標的を探します。
Telnetで開いているポートを検知すると、拡張モジュールはMiraiとほぼ同じ方法でブルートフォース（総当たり）攻撃によるシェルログインを行い、標的に入り込もうとします。この際、HajimeはMiraiが備えている61個の工場出荷時パスワードのデフォルトパスワードのリストを使用し、Atheros社製の無線LANルーターおよびアクセスポイントの工場出荷時設定である「root/5up」と「Admin/5up」の2つの新たなエントリーを追加します。さらに、Hajimeは初期設定のシード値を設定したパスワードオブザデイ「バックドア」を使用して、ARRIS社製モデムに侵入できます。

Hajimeは、認証情報の固定シーケンスに従うことはありません。ラドウェアでハニーポットログを確認したところ、悪用（エクスプロイト）の際に使用する認証情報は、標的のログインバナーによって変化すると考えられます。そうすることで、限られた試行回数内でデバイスに巧みに付け入る確率を高め、結果的にロックされているシステムアカウントや、デバイスによって所定の期間ブラックリストに掲載されているIPを回避します。

Hajimeの実行段階では、MiraiのようなIoTボットが悪用すると思われているポートをフィルタリングして、デバイスへのさらなるアクセスを防ぎます。

- **TCP/23 (Telnet)** - Miraiおよび大半のIoTボットネットの主要なエクスプロイトベクトル
- **TCP/7547 (TR-069)** – Miraiの亜種がDT攻撃で最初に使用
- **TCP/5555 (TR-069)** - TR-069で一般に使用される代替ポート
- **TCP/5358 (WSDAPI)** – WSDAPI（Web Service on Devices API）は、マイクロソフトが提供する組み込みデバイス用のオープンなDPWS（Device Profile for Web Services）仕様を相互利用可能にする実装用APIです。

8 https://securelist.com/blog/research/78160/hajime-the-mysterious-evolving-botnet/

同時に、Hajimeは「CWMP_CR」の名前が付いた既存のファイアウォールルールの削除も試みます。CWMPとは、CPE WAN管理プロトコル（TR-069）のことです。特定のIPやサブネットの管理を行うために、ISPが設定したCWMPのルールを削除すると、ISPはCPEデバイスの制御ができなくなってしまいます。

デバイス制御をロックするほかに、HajimeはUDP/1457ポート、およびUDPとTCP用にランダムな大きい番号のポート（1024以上）を開きます。その際に、UDP/1457ポートでBitTorrent DHTおよびuTPを使用できるようにして、自分のピアツーピアCnCネットワークを構築します。ランダムな大きい番号のポートは、感染プロセスで使用するローダーサービスのために役目を果たし、新たな標的にマルウェアをリモートでダウンロードします。

Hajimeは作業ディレクトリとして揮発型のファイルシステムを使用することを好み、デバイスを再起動すると確実に不正アクセスの痕跡をすべて消します。Hajimeは持続型ではなく、つまりデバイスを再起動することで感染を除去することができます。ただし、それは次に感染するまでの話です。

手法は理解できても、理由は不明？

ラドウェアではHajimeの仕組みについては説明してきました。ただし「なぜ」：Hajimeの目的は何なのか？は依然謎です。特に、Hajimeに起因する攻撃はまったくありませんでしたし、そのためのペイロードも携えていません。それにもかかわらず、Hajimeは洗練されている上にうまく設計されており、あっという間に別目的で使用できるほど高い柔軟性を持っています。

Hajimeの作者、そしてその意図や目的が「グレイ」であることについては数多くの臆測を呼んできました。本来の作者の動機を知るよしはありませんが、このボットネットは本来のオーナーからハイジャック（奪取）できると考えられます。Rapidity Networks社のEdwards氏とProfetis氏 が Hajimeの当初の暗号化実装にある脆弱性を発見し、そのメッセージングプロトコルを書き換えることができたため、その複雑なマルウェアには脆弱性がないというわけではないことが判明しました。先述の通り、脆弱性はパッチが適用され更新されましたが、柔軟なバックエンドを持ち、犯罪行為に利用される可能性が高いこの規模のボットネットは、間違いなくブラックハット（悪意のあるハッカー）の注目を集めるでしょう。つまり、ボットネットへの「秘密鍵」を持つ人は誰でも、その運命のカギを握っていることになるのです。

柔軟で高い拡張性を有することから、Hajimeは容易に別の目的で用いられ、次にあげるようなものを実行するために利用されます。

- **DDoS攻撃** いくつかのDDoS攻撃を実行したコードを備えているMiraiとほぼ同じ方法で、コマンドでDDoS攻撃の破壊を行う同様以上の攻撃ベクトルを備えたHajimeの拡張モジュールを、容易に作ることができます。
- **大規模に分散した脆弱性のスキャニング** これにより、新たな脆弱性の発見後数時間以内に、ハッカーは脆弱で無防備な公的サービスを検知し、それらを悪用できるようになります。（これまでに分かっているとおり、大半のシステムには数時間ではパッチが適用されません）カスタムエクスプロイトモジュールは、対応するプラットフォームの1つにバイナリでコンパイルできるならば、あらゆる言語で記述でき、トレントオーバーレイを通じて配信され、インターネット上の数万どころか、数十万もの分散したノードで実行されることになります。
- **大規模な監視ネットワーク** 拡張モジュールがカメラのRTSPストリーミングを利用する可能性があります。
- **IoT Brickerネットワーク** BrickerBotの動作を利用して、atkプログラムに小さく簡単な変更を加え、CnCチャネルを通して「Plan B」コマンドを受信する際に、自己破壊シーケンスを実行します。ハッカーは、地理的IPに基づいた地域や都市に応じて、感染したデバイスをすべて操作不能にして、秘密裏に特定の地域や都市を標的にして狙うことができます。

現在のところ、Hajimeは本来の作者の制御下にあると考えられ、大半の部分において、その意図は正しいものだと思われます。しかし、この噂の救世主（ホワイトナイト）が次の標的候補を積極的に探してスキャニングし、そしてすべての標的を人質にし、ボットネットを拡大し続けている理由については疑問が残ります。その意図が正しいものであるならば、なぜCWMPのルールをそのままにしておかないのでしょうか？なぜISPが不適切に処理したものを改善しないのでしょうか？なぜファイアウォールのルールをそのままにしておかない、または、なぜそれらを変わりやすくしたままでデバイスを解放するのでしょうか？その代わり、Hajimeは再起動されるまでデバイスを人質にしたままです。

HajimeがIoTボットネットの未来をわずかながら見せてくれているとすれば、IoT業界が既存製品と新規製品の保護を検討し始めることを祈りたいと思います。そうでなければ、私たちのこれからの望みと未来は、BrickerBotで名声をあげたJanit0rのような、グレイハット自警団を用いる困難な方法で脅威を一掃することにかかっているかもしれません。

BRICKERBOT: 病よりも深刻な治療？

2017年4月5日、ラドウェアは驚くような発見を発表しました。4日間にわたり、当社のハニーポットが、世界中のいくつかの場所から実行された1,895件の永続的なサービス拒否（PDoS：Permanent Denial-of-Service）の攻撃試行を記録したのです。攻撃の唯一の目的は、IoTデバイスに不正アクセスし、そのストレージを破壊することでした。この攻撃の猛烈さに加えて、ラドウェアのハニーポットでは短命攻撃（BrickerBot.1）のわずかな時間での試行と、同日にPDoS攻撃試行を始めたそれに非常に類似したボット（BrickerBot.2）を記録しました。いずれのボットも1時間違わずに発見され、2番目のものは強度がやや低いものの、徹底度合いが高まっており、そのロケーションはTOR出口ノードで隠されていました。

デバイスへの不正アクセス

BrickerBotのPDoS攻撃では、Telnetブルートフォース（MiraiやHajimeが使用したものと同じエクスプロイトベクトル）攻撃を使用して、標的のデバイスに侵入します。BrickerBotでは、バイナリをロードしようとはしません。作者であるJanit0rによると、むしろよく知られたポートをスキャンし、破壊的ではない方法でデバイスの保護を試みます。これに失敗すると、BrickerBotは Janit0rが言う「Plan B」に戻ります。この「B」は、「Brick（操作不能にするもの）」です。

デバイスの破壊

デバイスへのアクセスが成功すると、同時にPDoSボットは最終的にはストレージの破壊を引き起こす一連のLinuxコマンドを実行し、インターネット接続やデバイス性能を妨害してデバイス上のすべてのファイルを消去するコマンドを実行します。図11を参照。

```
fdisk -l
busybox cat /dev/urandom >/dev/mtdblock0 &
busybox cat /dev/urandom >/dev/sda &
busybox cat /dev/urandom >/dev/mtdblock10 &
busybox cat /dev/urandom >/dev/mmc0 &
busybox cat /dev/urandom >/dev/sdb &
busybox cat /dev/urandom >/dev/ram0 &
fdisk -C 1 -H 1 -S 1 /dev/mtd0
w
fdisk -C 1 -H 1 -S 1 /dev/mtd1
w
fdisk -C 1 -H 1 -S 1 /dev/sda
w
fdisk -C 1 -H 1 -S 1 /dev/mtdblock0
w
route del default;iproute del default;ip route del default;rm -rf /* 2>/dev/null &
sysctl -w net.ipv4.tcp_timestamps=0;sysctl -w kernel.threads-max=1
halt -n -f
reboot
```

図11. BrickerBot.1のコマンドシーケンス

標的となった特有のデバイスは次のとおりです。

- **/dev/mtd – メモリテクノロジーデバイス** – フラッシュの特性に一致する特有のデバイスタイプ
- **/dev/mmc – マルチメディアカード** - メモリカードスタンダード、ソリッドステート・ストレージ・メディアに一致する特有のデバイスタイプ

標的

「Busybox」コマンドをMTDおよびMMCの特有デバイスと組み合わせて使用しているため、この攻撃は、Telnetポートを開き、それをインターネットに開示したLinux/BusyBoxベースのIoTデバイスを明確に標的としていることになります。これらは、Mirai、Hajime、またはそれに関連したIoTボットネットが標的とするデバイスと一致します。

限られた数のIPアドレスから発生したPDoS攻撃の試行は、世界中に広がります。攻撃されるすべてのデバイスではポート22（SSH）が暴露され、Dropbear SSHサーバーの旧バージョンが稼働し、古いファームウェアを使用しています。デバイスの大半は、Shodanで無線CPEデバイス、無線アクセスポイントや無線ブリッジ（ビーム指向性機能あり）と識別されています。

BRICKERBOT.3: JANIT0Rが復讐のためにカムバック

2017年5月、ラドウェアは別の場所にあるハニーポットで、新たなコマンドシーケンスを備えた新たなバージョンのBrickerBotのPDoS攻撃（BrickerBot.3）を発見しました。図12を参照。BrickerBot.3の最初の12時間についてのレポートを仕上げてから1時間後、BrickerBotの作者とみられる人の記事とコメントが、Catalin Cimpanu氏のブログで公開され、そこで現在まで200万のデバイスを使用不能にしたと主張したのです。

最初の12時間

攻撃の最初の12時間、総数1,118にのぼるPDoS攻撃の試行を記録しました。攻撃はすべて、限定された数のクリアネットのIPアドレスから発生しています。
攻撃のソースIPに基づくZoomEyeとShodanでの検索では、そのすべてがDropbear SSHサーバーの古いバージョンで稼働していることが分かりました。

攻撃は4月21日の12時（グリニッジ標準時）に始まり、最初の12時間で攻撃を実行するボット数は15にまで増加しました。図13を参照。

当社のハニーポット上でPDoS攻撃を実行するために使用しているデバイスは、BrickerBot.1の（攻撃対象）デバイスとは一致していません。また、BrickerBot.1は、限定された数のクラリネット接続デバイスを悪用して攻撃を実行しますが、2つの間に直接の相関関係はありません。[10]

痛いところを突く: ISPのSIERRA社の場合

2017年4月10日、カリフォルニア州の小規模ISPであるSierra Tel社の顧客が、インターネットと電話サービスが途絶するトラブルに見舞われました。モデムの問題だと判断したSierra社の当初の対応では、顧客のモデムZyxel HN-51を持ち込んでもらい、新たなものに取り替えていました。

しかし、誤作動するモデムの数は急増し、新たなデバイスの提供は（在庫がなくなり）できなくなりました。その間、Janit0rはBleeping ComputerのジャーナリストであるCatalin Cimpany氏に接触し、この事件への関与を主張しましたが、その時点では全国ニュースには至りませんでした。Sierra Tel社が完全にこの問題に対処し、すべての顧客サービスを復旧させるまでには2週間を要しました。[9]

Bleeping Computerの記事には次のように書かれています。
「Mirai、BrickerBot、Hajime、Wifatch、Gafgyt、Imeijなどのさまざまな種類のIoTマルウェア間で繰り広げられる目に見えない縄張り争いの結果、Sierra Tel社のZyxelモデムがオフラインになってしまうことは十分あり得ます。IoTマルウェアファミリーが増えれば増えるほど、Sierra Tel社事件のような問題をさらに引き起こすことになるでしょう。」

```
busybox cat /dev/urandom >/dev/mtdblock0 &
busybox cat /dev/urandom >/dev/sda &
busybox cat /dev/urandom >/dev/mtdblock10 &
busybox cat /dev/urandom >/dev/mmc0 &
busybox cat /dev/urandom >/dev/sdb &
busybox cat /dev/urandom >/dev/ram0 &
busybox cat /dev/urandom >/dev/mtd0 &
busybox cat /dev/urandom >/dev/mtd1 &
busybox cat /dev/urandom >/dev/mtdblock1 &
busybox cat /dev/urandom >/dev/mtdblock2 &
busybox cat /dev/urandom >/dev/mtdblock3 &
fdisk -C 1 -H 1 -S 1 /dev/mtd0
w
fdisk -C 1 -H 1 -S 1 /dev/mtd1
w
fdisk -C 1 -H 1 -S 1 /dev/sda
w
fdisk -C 1 -H 1 -S 1 /dev/mtdblock0
w
route del default;iproute del default;ip route del default;rm -rf /* 2>/dev/null &
sysctl -w net.ipv4.tcp_timestamps=0;sysctl -w kernel.threads-max=1
halt -n -f
reboot
```

図12. BrickerBot 3

9 https://www.bleepingcomputer.com/news/security/us-isp-goes-down-as-two-malware-families-go-to-war-over-its-modems/

10 情報を正確かつ完全に示すとすれば、攻撃はBrickerBot.1とBrickerBot.2攻撃を検知したものとは異なるハニーポットで検知されました。

攻撃を仕掛けるデバイスは世界中に広がり、特定の地域に集中することはありません。BrickerBot.1のソースの場所にも相関関係がありません。図14を参照。

BrickerBot.1とBrickerBot.2に倣って、このボットもMiraiエクスプロイトベクトルを使用して標的に不正アクセスを行います。Telnetのポートが公知となり、工場出荷時の認証情報を変更していないBusyboxベースのLinuxデバイスは、すべて標的となる可能性があります。

BRICKERBOT.4

午後5:22から午後8:44（グリニッジ標準時）の間、同じハニーポットが非常によく似たまったく別のコマンドシーケンスを検知しました。この攻撃は、クリアネットにある1つのデバイスのみから実行されました。調査の結果、Dropbear SSHサーバーの旧バージョンであることが分かりました。この孤立したボットは90回攻撃を行い、その後再び検知されることはありませんでした。

Bricker.Bot3と比較して、BrickerBot.4のコマンドシーケンスは、破壊しようとするブロックデバイス数が少数です。図15を参照。

JANIT0R: 救世主なのか？

BrickerBot.3とBrickerBot.4を発見した数時間後、Catalin Cimpanu氏はBrickerBotの作者について記事を発表しました。そこで、HackForums.netにおいて「Janit0r」の名前で知られている人物が、BrickerBotの作者であることが示唆されました。Janit0rとの議論に基づくと、BrickerBotは私たちが当初考えていたよりも、さらに複雑でおそらくはるかに大規模です。

「他の多くの方々と同じく、私は2016年のIoTボットネットによる無差別なDDoS攻撃には驚かされました。大規模な攻撃は、最終的には業界で足並みを揃えた行動を強制させるものだと私は思い込んでいましたが、この記録破りな攻撃の数ヶ月後には、誠実な努力にもかかわらず、従来の手法ではこの問題を迅速には解決できないことが明らかになりました。私は、自分のプロジェクトを『インターネット化学療法』の形態だと考えています。そして自分自身のことを、冗談ながら『医師』と考えることがあります。化学療法は、正気であれば誰も健康な患者に施すことのない厳しい処置ですが、インターネットは2016年第3四半期と第4四半期には深刻な病となり、並の治療では効果がありませんでした。」

Janit0rは、2016年11月以降、200万（以上）のデバイスを使用不能にしたと主張しています。

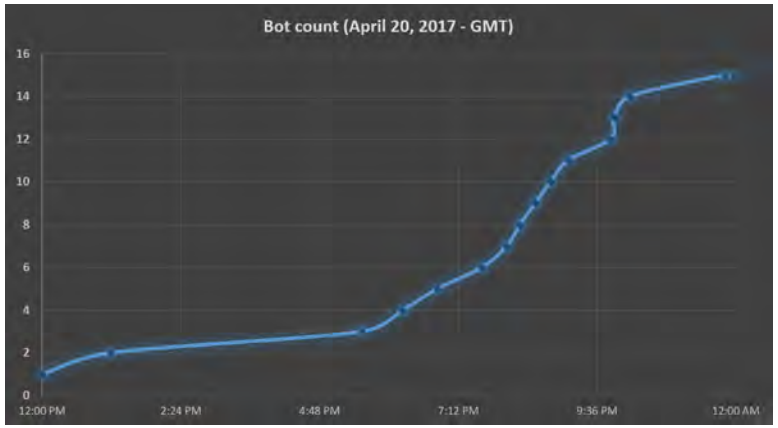


図13. BrickerBot.3の時系列に見た拡大状況

図14. BrickerBot.2が攻撃実施に使用したデバイスの地理的分布

```
1  fdisk -l
2  busybox cat /dev/urandom >/dev/mtdblock0 &
3  busybox cat /dev/urandom >/dev/sda &
4  busybox cat /dev/urandom >/dev/mtdblock10 &
5  busybox cat /dev/urandom >/dev/mmc0 &
6  busybox cat /dev/urandom >/dev/sdb &
7  busybox cat /dev/urandom >/dev/ram0 &
8  fdisk -C 1 -H 1 -S 1 /dev/mtd0
9  w
10 fdisk -C 1 -H 1 -S 1 /dev/mtd1
11 w
12 fdisk -C 1 -H 1 -S 1 /dev/sda
13 w
14 fdisk -C 1 -H 1 -S 1 /dev/mtdblock0
15 w
16 route del default;iproute del default;ip route del default;rm -rf /* 2>/dev/null &
17 sysctl -w net.ipv4.tcp_timestamps=0;sysctl -w kernel.threads-max=1
18 halt -n -f
19 reboot
```

図15: BrickerBot.4のコマンドシーケンス

当局とハードウェアベンダーが、IoTセキュリティの状態を改善させるために決定的行動を取るまでは、しばらくの間、Janit0rがBrickerBot攻撃で揺さぶり続けそうです。

BrickerBot.1と同様に、BrickerBot.3とBrickerBot.4はBusyboxベースのLinuxデバイス、主としてIPカメラやDVRなどのIoTデバイスを標的とします。2017年4月13日のラドウェアのBrickerBotに関するブログに記述があるとおり、ラドウェアではBrickerBot.1のコマンドシーケンスをテストし、実在するデバイスへの影響を評価しました。Sricam社製AP003メタルガン型防水屋外用弾丸IPカメラは、Mirai「対応」として知られています。

名称からは堅牢さがうかがえますが、BrickerBot.1からコマンドシーケンスを実行すると、カメラはネットワークから切断され、再起動によって反応しなくなりました。カメラにある特殊ボタンで工場設定をリセットしても、IoTデバイスは復旧できませんでした。効果的に使用不能とされました。

図16: JanitOrを含む会話の引用

悪い知らせは、BrickerBotは極めて効果的であると言うことです。良い知らせは、それは以前感染したIoTデバイスでのみ動作すると言うことが分かったことです。

BRICKERBOTの仕組みを理解する

ラドウェアの分析で、BrickerBotの標的の特定方法とターゲティング方法が明らかになりました。攻撃の前には、少なくとも攻撃の間に使用される同じデバイスからは、アクティブなスキャニングの痕跡はありません。BrickerBotデータで、限定された数のデバイスのみが攻撃を仕掛けたことが分かりました。その上に、単純に姿を消す前に、大半が90回攻撃を仕掛けました。このマジックナンバー90は、2回目にBrickerBot攻撃が高まった際に、さらに一層明確になりました。

1つ大きな問題が残っています。BrickerBotはどうやって感染しているデバイスを検知するのでしょうか？ラドウェアでは、すぐに分かりました。答えは常に目の前にあります。ラドウェアのハニーポットは、既知のエクスプロイトポートで待機するよう設計されています。BrickerBotが同じことをしたらどうでしょうか？Miraiの亜種やHajimeなどのボットネットによるエクスプロイト攻撃試行を、受動的に検知するとしたらどうでしょうか？インターネットをアクティブにスキャンして新たな標的を見つけるのではなく、感染したIoTデバイスをスキャンするために、ポートTCP/23（telnet）とポートTCP/7457（TR069）でまさに待機するだけです。確かに、この技術は個々のデバイスからインターネットの大部分をアクティブにスキャンするよりも、こっそり行える上にさらに効率的です。

直近に発見されたBrickerBotのソースIPアドレスの1つを使用して、ラドウェアではポートTCP/23のTCP接続テストを行いました。接続は確立でき、ポートは即座にサーバーがクローズしました。数秒の間に、同じインターネット接続で展開しているハニーポットは、私たちが呼び出した同じBrickerBotのソースIPからBrickerBotシーケンスが送られてくることを示し始めました。同じBrickerBotは、攻撃試行数が正確に90回になるまで攻撃を続け、その後去りました。

さらに、以前攻撃が高まった際にBrickerBotの感染を受けたデバイスをいくつかテストしてみると、さらに多くのポートが開いていることが分かりました。ポート7547および19058へのTelnet接続は、絶えずBrickerBot攻撃の引き金となっており、ボットはIoTボットエクスプロイトが使用する大半の既知のポートで待機していることが示されています。その後のBrickerBot攻撃試行において、ラドウェアはやや異なるシーケンスに気が付きました。これは、当社のハニーポットコード、またはJanit0rが改良したBrickerBotコードに行った変更点に関係するかもしれません。

図17. BrickerBotの地理的な分布（1ドット = 1ボット、合計223ボット）

BRICKERBOT: 最終結論

BrickerBotのボットネットは世界中に拡大しました。ラドウェアでは223のアクティブノードを特定しました。図17を参照。Shodan.ioおよびzoomeye.orgクエリを使用すると、ほぼすべてのデバイスが旧バージョンのDropbear SSHサーバーで稼働していることが分かっています。大半が古いファームウェアのままのようで、ホスト名が『HACKED-ROUTER-HELP-SOS-DEFAULT-PASSWORD』や『HACKED-ROUTER-HELP-SOS-WAS-MFWORM-INFECTED』のものもあり、これらのデバイスは脆弱性があることが知られています。

世界規模の痕跡は別として、BrickerBotは大半のボットネットとはまったく異なっています。既に述べたように、一般的なボットネットは「大将（CnC）」の命令に対して邁進する軍隊のように機能します。一方、BrickerBotの個々のインスタンスは、感染したIoTデバイスに目覚めさせられた時のみ行動する孤独な眠れる熊のようなものです。いったんBrickerBot「熊」が目覚めると、感染したIoTデバイスを使用不能にするように90回の攻撃を仕掛けます。しかし、当社では不正アクセスされたIoTデバイスによってのみ目覚めさせられると考えており、通常は、Mirai、Hajime、またはモーニングコールとして機能する他のIoT感染ボットによる不正アクセスがその契機となります。

つまり、自分のデバイスをクリーンでセキュアにしておけば、ペーパーウェイトの大コレクションを収集することを不安がる必要はありません。

ボットの撃退

IoTの制御を巡り格闘している間にも...

MIRAI: 有害

- これまでで最も強力なボットネット
- 新たなレベルのDDoS攻撃
- 数Tbps攻撃の可能性も
- 単純で集結が容易な新たなボットがDDoS-as-a-Serviceエコノミーに影響をおよぼす

HAJIME: 有益（少なくとも現在のところは）

- 安全性に欠けるIotデバイスを人質に取り、DDoSボットネットに加わらないようにする
- 高機能 - IoTボットやボットネットの未来を垣間見ることが可能
- 積極的にスキャンし感染
- 更新と新たな拡張のためにCnCチャネルをオープンに
- 真の目的は謎のまま

BRICKERBOT: 自警団

- 安全性に欠けるIotデバイスを破壊し、DDoSボットネットに加わらないようにする
- 他のボットに不正アクセスされているデバイスのみを攻撃

…自分のIoTデバイスの守り方

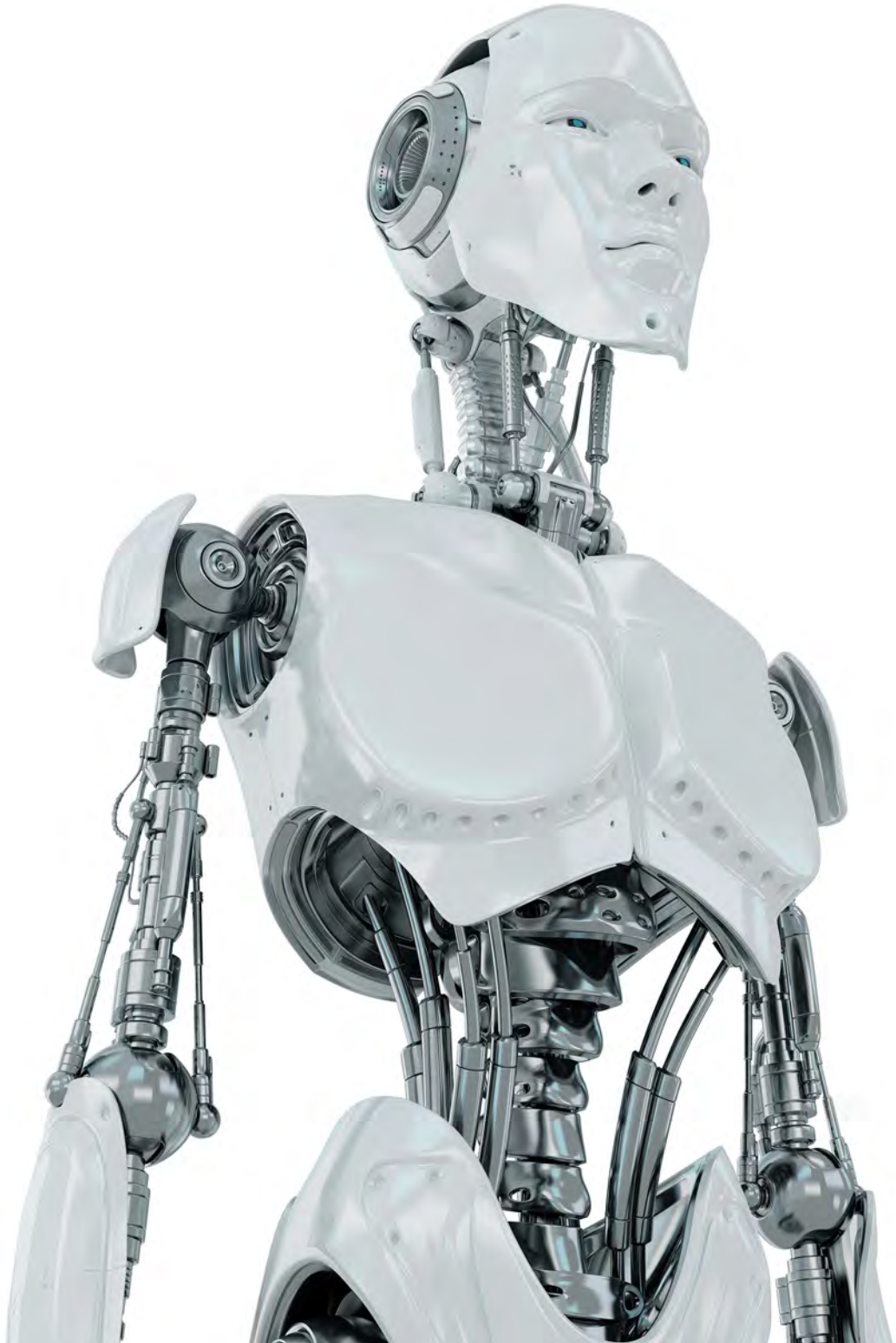
注意点:

- デバイスの工場出荷時の認証情報を変更する
- デバイスへのTelnetアクセスを無効にする
- 頻繁にファームウェアを確認し更新する
- 近い将来に発生する頻度の多い、大規模なDDoS攻撃に備えて引き締める
- 適切なインシデントレスポンス計画を策定し、訓練を実施して従業員をトレーニングし、計画の有効性をテスト・評価する
- ネットワーク振る舞い分析を使用してトラフィックの異常を検知し、その結果と自動シグネチャ生成を組み合わせて迅速に効果的な（攻撃）軽減を行う
- ユーザーおよび組織の振る舞い分析を使用し、早期にトラフィックの細かな異常を見極める

今何をすべきか？

ボットネット、Webスクレイピング、IoTゾンビの脅威の状勢は動的で、ますます複雑化しています。ソリューションは、軽減しなければならない問題と同じぐらい動的であり続ける必要があるため、固定的なソリューションのリストを出すことは裏目となります。

知識は力。情報を調べ、学び、そしてラドウェアとつながり続けることをお奨めします。当社によるBrickerBotの発見で証明されたとおり、新たに現れ、進化し続けるサイバーセキュリティの脅威に対して、ラドウェアは最先端の技術を備えています。



radware