# サイバー救急センターレポート

- 脅威管理とインシデント対応をする人へ -

第 2 号

2018 冬

CYBER EMERGENCY CENTER

2018 年 3 月 2 日

サイバー救急センター

# サイバー救急センターレポート

# 第 2 号 / 2018 冬

## 目　　次

# はじめに

内田 法道

株式会社ラック
サイバー救急センター長

昨年は、引き続き高度な標的型攻撃（APT: Advanced Persistent Threat）の相談があったことに加え、自己増殖型のランサムウェア（WannaCry 等）の出現、仮想通貨発掘ソフトウェアを不正侵入したインターネット公開サーバで実行するなど、金銭目的の攻撃者グループの活動も活発な年でした。

WannaCry の被害を見ていて感じたのは、海外では幾つか報告がありますが、企業を対象とした標的型のランサムウェアの可能性です。WannaCry の被害状況が私に示唆したものは、日本企業のオフィスで利用している PC は適切にアップデートが実施されているものの、①社内サーバのアップデートは徹底されていない、②制御系のネットワークとオフィス系のネットワークが意外につながっている、③制御系の機械を管理・監視する機器に Windows OS が利用されている、の 3 点でした。

これに、APT で利用されている遠隔操作技術が組み合わさると、企業の重要な社内サーバや制御系で利用している Windows 機器を暗号化等で利用できなくし、脅迫するという脅威が想起されます。業務が 1 日止まると数億円の損害が発生する製造ラインの機器、1 日に数億円売上げる EC サイトの重要サーバ等が停止して、妥当な額の身代金が要求された場合、企業は難しい選択を迫られます。

上記を含め、新たな脅威が発生した際には、世間や専門家の騒ぎに惑わされずに、これまで想定外と思っていた脅威リストを冷静に再点検し、必要な脅威への予防策や事後策を検討しましょう。

# サイバー119 の出動傾向

## 2017 年 10 月～12 月の出動傾向

この期間も、不正送金マルウェアに感染させるばらまき型のメールによるマルウェア感染の相談はきておりますが、各企業の仕事納めであった 12 月 27 日～28 日にかけて、そのようなメールが多数ばらまかれ、誤って添付ファイルを開いてしまった企業からの問い合わせがありました。また、年始の仕事始めに、社員や職員が添付ファイルを開いてしまっている可能性もあるため、注意ください。

インターネットに公開しているサーバを対象とした不正侵入のインシデントが増加傾向にあります。不正侵入後は、Monero 等の仮想通貨を発掘するソフトウェア（コインマイナー）がインストールされることが多く、金銭目的の攻撃者グループによるものと考えられます。彼らの狙いは、サーバの保持している「情報」ではなく「演算能力」ですので、管理に不備があるサーバを見つけて侵入してきます。アップデートの未適用、ファイアウォールの設定ミスなど、管理の不備に気づかずに運用しているサーバがないか、改めて確認ください。

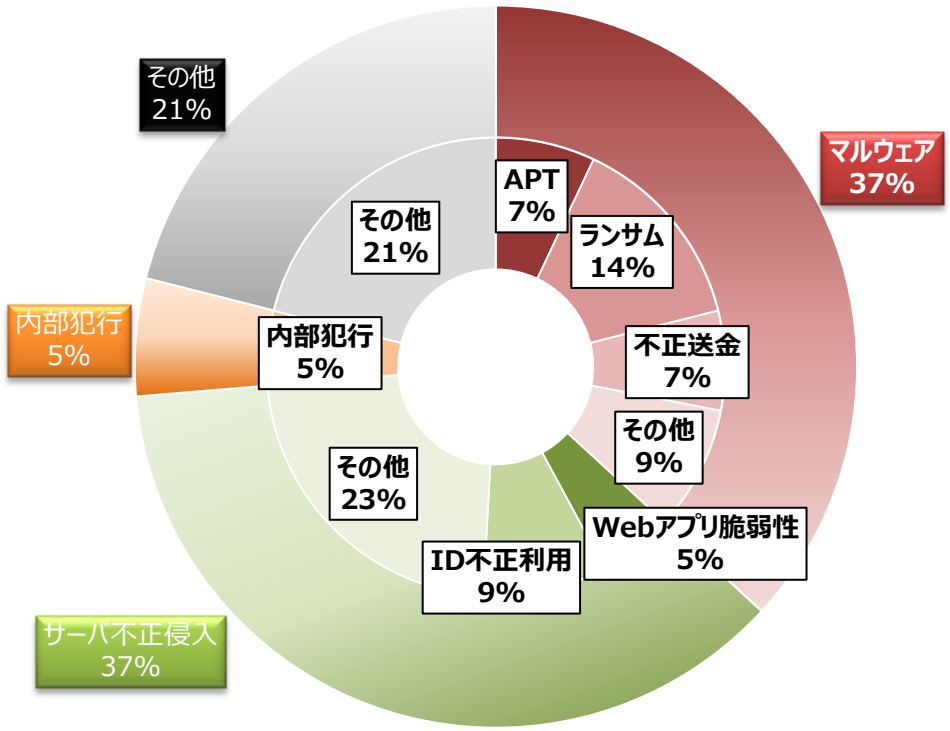


**図 1-1　2017 年 10 月～12 月のインシデント傾向**

# マルウェア関連のインシデント傾向と対策

## (1)APT攻撃（Advanced Persistent Threat攻撃、標的型攻撃）

APT攻撃と推測されるインシデントの相談を複数受けています。被害組織の業種、攻撃手法、利用されているマルウェア等は様々ですが、Menupass(APT10)、AuroraPanda(APT17)、WinttiおよびDragon OKの攻撃グループが関わっている可能性が疑われるインシデントも確認しています。

確認されたインシデントの大半は、1～2 年前から不正な侵害を受けており、プロキシログの定期的なチェックの結果や、次世代ファイアーウォールでの検知等から、インシデントが発覚しています。

インシデント発覚までに時間が経過すればするほど様々な痕跡が失われることから、情報漏えい等の被害範囲を特定することは困難になります。インシデントの早期発見に向けたセキュリティ対策の検討は必要ですが、重要情報のセグメント分離といった被害範囲を狭める対策も検討ください。

## (2)ばらまき型メール（ランサムウェア、不正送金マルウェア）

ばらまき型メールによるランサムウェア感染（ファイル暗号化被害）や、不正送金系マルウェア感染が継続して発生しています。

ばらまき型メールでは、Microsoft Office でアプリケーション同士のデータ転送に利用されているDynamic Data Exchange (DDE)の機能を悪用してマルウェアに感染させる手法を確認しております。この機能を悪用したOfficeファイルは、通常の拡張子（.docx、.xlsx など）のため社員や職員が不安に思わず開いてしまう可能性がありますので、組織内への注意喚起等を検討ください。

なお、JC3、警視庁、警察庁といったところから、ウイルスメールに関する注意喚起の情報が定期的に発信されていますので、見覚えのないメールが来た際には、すぐに開封するのではなく、注意喚起情報に当該メールの情報が掲載されていないか確認するようにしてください。

■ウイルスメール注意喚起情報（JC3）
https://www.jc3.or.jp/topics/virusmail.html
■警察庁 Twitter
https://twitter.com/npa_koho
■警視庁 Twitter
https://twitter.com/mpd_cybersec

# 公開サーバ関連のインシデント傾向と対策

2017 年 7 月～9 月の期間から継続して、公開サーバの脆弱性を悪用して不正侵入した攻撃者が、仮想通貨を発掘するマルウェア（コインマイナー）を設置するインシデントの相談を複数受けていますが、特に 12 月下旬には、Oracle 社の WebLogic Server を利用しているサーバに対する不正侵入インシデントの相談が増加しています。

WebLogic Server における任意コード実行の脆弱性(CVE-2017-10271)については、脆弱性の公表は 10 月でしたが、12 月下旬に当該脆弱性を悪用する攻撃コードが公開されたことから、脆弱性対応が実施されていなかった組織のサーバにおいて被害が多発したと考えられます。

脆弱性の公表から不正侵入に悪用されるまでに、約 2 か月の期間があったことから、日々脆弱性情報をチェックし、パッチの適用を実施する脆弱性管理が適切に運用されていれば、防ぐことができたインシデントと考えられます。WebLogic Server の脆弱性を悪用した不正侵入は、しばらく継続して発生する可能性がありますので、これを機会に

- 公開サーバの棚卸し
- 脆弱性対応状況の確認
- パッチやアップデートの適用

を実施することを推奨します。

コインマイナーを設置する攻撃者グループの狙いは、組織が保有している情報ではなく、サーバの演算能力と考えられます。仮想通貨の普及、レートの上昇、匿名性の高い仮想通貨の登場など、外部環境の変化により、この傾向は継続する可能性がありますので、公開サーバの脆弱性管理には注意ください。

# 2017 年の出動傾向

2017 年の総括として、過去 2 年の出動傾向と比較すると、以下のような傾向が確認されました。

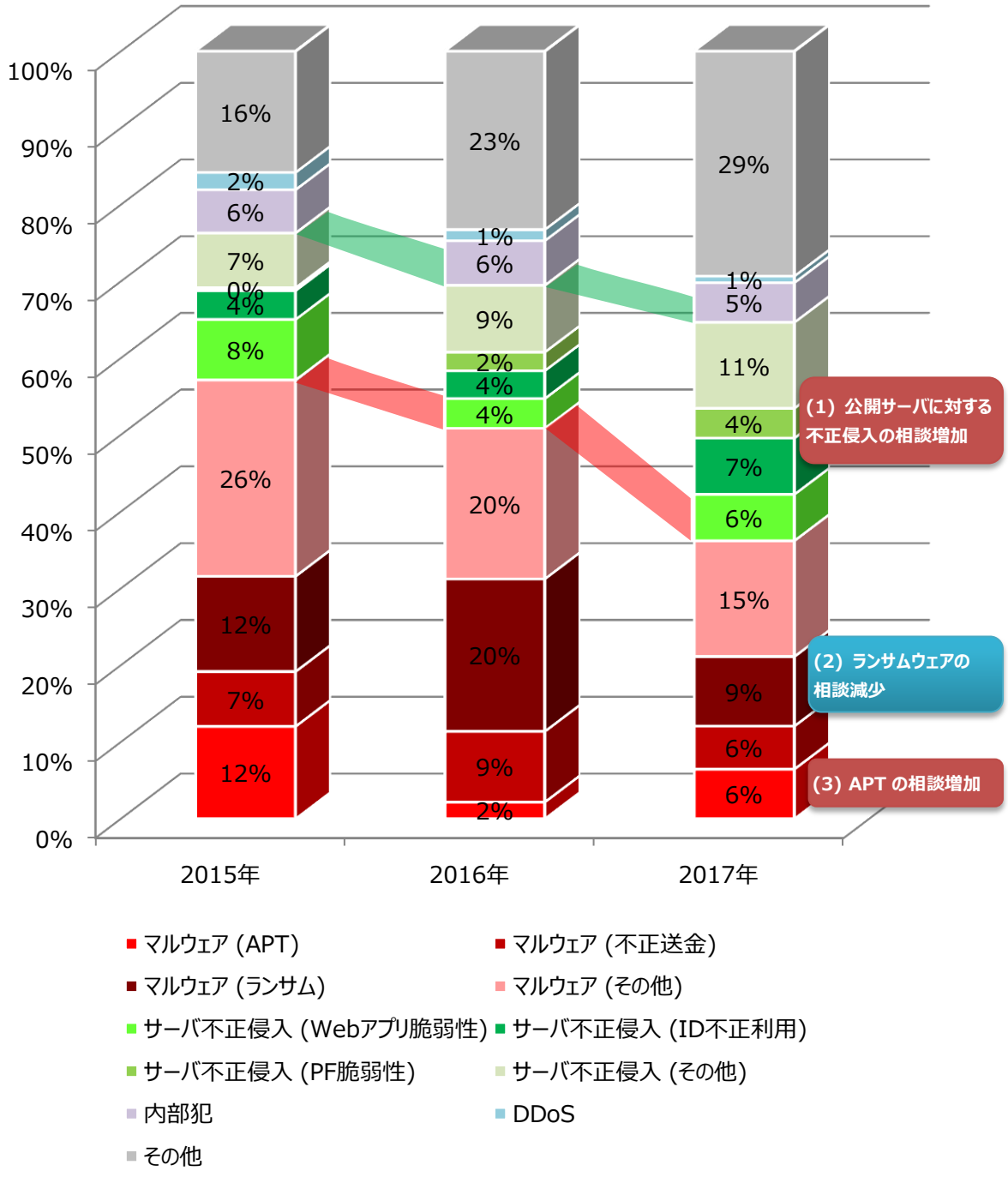


**図 1-2　2015 年～2017 年のインシデント傾向**

### (1)公開サーバに対する不正侵入の相談増加

脆弱性を悪用した公開サーバに対する不正侵入事案としては、3 月に Apatch Struts2、12 月に WebLogic の脆弱性を悪用したインシデントの相談を複数受けた結果、過去 2 年と比べて 2017 年は公開サーバに関連する出動の割合が増加しました。

脆弱性対策としてアップデートやパッチを適用することが重要ですが、アップデートやパッチを適用するにあたって検証に時間を要するいったケースが多く見られました。脆弱性の対策に時間がかかるほど、不正侵入を受ける確率が高くなることから、迅速なアップデートやパッチ適用に向けた運用体制の準備を行う必要があります。

また、ファイアウォールの設定不備がある公開サーバや CMS のコントロールパネル経由での不正侵入インシデントも多発しています。いずれも、単純なパスワードを設定したアカウントが悪用されていますので、強固なパスワード設定や不要なアカウントの削除といった対策を行う必要があります。

### (2)ランサムウェアの相談減少

ランサムウェアとしては、5 月の Wannacry の脅威がありましたが、年間を通してみると、サイバー救急センターへのランサムウェアに関する相談の割合は減少しました。多くの組織で、ランサムウェアに関するインシデント対応が可能になり、サイバー救急センターへの相談が減少したのではないかと、推察されます。

しかしながら、ばらまき型メール自体は減少していないことから、引き続きばらまき型メールに対しては注意ください。

### (3)APT 攻撃（標的型攻撃）の相談増加

2015 年は、「Emdivi」と呼ばれる遠隔操作マルウェアを用いた日本への APT 攻撃が多発したのは、まだ記憶に残っていると思います。2016 年も APT 攻撃による個人情報漏えいの事件がメディア等で取り上げられました。

一方で、2017 年は APT 攻撃による被害がメディアで大きくとりあげられることはありませんでしたが、サイバー救急センターへの APT 攻撃の相談は増加しています。2017 年も、APT の攻撃者グループは、日本企業が保有する技術情報等を狙っており、直接対象となる企業を狙うだけでなく、取引先や関連会社も標的に含まれている可能性があります。

「自組織の保有情報は攻撃者に狙われている」という意識を常に持つだけでなく、自組織の情報を保有している取引先や関連会社が狙われているリスクも含めて対策を検討ください。

# 攻撃者の残した痕跡に学ぶ

## プロキシサーバのすゝめ

サイバー救急センターで相談を受けるインシデントの約半数は、マルウェア感染に関連するものです。マルウェアの多くは、インターネット上に攻撃者が用意した C2（コマンド＆コントロール）サーバと通信して、外部からの指令を受け取って動作したり、情報を外部に送信します。したがって、組織内部からインターネットへ出て行く通信（アウトバウンド通信）の監視や分析を行うことで、組織内部のマルウェア感染にいち早く気がつき、被害の発生や拡大を防ぐことが可能です。

一般的な組織では、アウトバウンド通信として Web（HTTP/HTTPS）通信は許可されているため、大半のマルウェアは C2 サーバとの通信に Web 通信を使用します。多くの組織では Web 通信の制御のために、Web プロキシサーバを導入しているものの、Web 閲覧を制限する URL フィルタリングを目的としており、残念ながらマルウェア対策としてはあまり活用されていません。

そのため、サイバー救急センターで対応したマルウェア感染のインシデントの経験をもとに、Web プロキシサーバを徹底的に活用する原則を幾つか紹介します。

### (1) アウトバウンド通信の原則

最初に組織内部からインターネットに対するアウトバウンド通信の原則を紹介します。

**① 必要なアウトバウンド通信のみを許可し、許可した通信はゲートウェイ機器を通過させる**

まずファイアウォールで全てのアウトバウンド通信を制限し、必要なアウトバウンド通信を洗い出して、必要な通信についてはゲートウェイ機器を通過させます（例：Web 通信の場合は Web プロキシサーバ、DNS 通信の場合は DNS サーバ）。これは「関所」となるゲートウェイを通すことで、アウトバウンド通信の細かな制御を行うだけでなく、ゲートウェイに残る「記録」をもとに通信の監視・分析を実現するためです。

アウトバウンド通信の制限で注意する点としては、まず DMZ 上のサーバからインターネットへの通信があります。DMZ 上のサーバからインターネットへの通信制限が行われておらず、ゲートウェイを経由せずに通信が可能な構成が多く見受けられます。次に、クラウドサービスの仕様の壁があります。ゲートウェイはクラウドサービスを利用することも可能ですが、必要なときにログの参照やダウンロードができないサービスもあるため、仕様を確認しておきましょう。万が一、ゲートウェイが侵害されると、MiTM（Man-in-The-

Middle 攻撃）に悪用される恐れがあるため、ゲートウェイのセキュリティ対策にも注意が必要です。

**② アウトバウンド通信は許可と拒否、双方のログを取得する**

マルウェア感染のインシデントでは、アウトバウンド通信のログを調査してマルウェアに感染した端末を特定することがあります。そのため、通信を許可したログを取得していないと、C2 サーバと通信している端末を特定できません。また、通信を拒否したログについても、C2 サーバとの通信試行のログが取得されます。そのため、インシデントに備えて許可および拒否、双方のログを取得します。

**③ 専用のゲートウェイを通過できないアウトバウンド通信を許可する場合は、通信元および通信先の双方を限定して許可する**

業務上の都合やソフトウェアの仕様によってはゲートウェイを通過させずに、アウトバウンド通信を許可する必要があります。その場合、必ず通信先は限定し、可能な限り通信元も限定した上で許可します。

次世代型ファイアウォールを利用している場合、通信元や通信先だけでなく、通信しているアプリケーションでも制限することができるため、必要に応じて次世代型ファイアウォールの機能を活用してください。

## (2) Web プロキシの原則

次に Web プロキシの原則を紹介します。

**① 可能な限り明示型のプロキシ構成とする**

Web プロキシサーバの構成には、明示型と透過型の構成があります。明示型の構成では、クライアント（OS やブラウザ等）に、Web プロキシサーバを明示的に指定して、当該サーバを経由して通信します。透過型の構成では、明示的な Web プロキシサーバの設定は不要で、通信経路上の機器で強制的に Web プロキシサーバを経由させます。

明示的な構成では、暗号化された HTTPS 通信の場合に CONNECT メソッドが使用されるため、以下のように通信先のドメイン名が記録されます。もちろん、CONNECT メソッド以降の通信は暗号通信となるため確認できません。一方、透過型の構成では CONNECT メソッドが使用されないため、接続先のドメイン名が正確に記録されないことがあるため、Web プロキシは明示型の構成が望ましいです。

```
CONNECT outlook.office365.com:443 HTTP/1.1
```

**図 2-1 明示型な Web プロキシサーバでの HTTPS 通信のログの例**

**② HTTPS は可能であれば復号し、可能であれば IP アドレスでの HTTPS 接続を拒否する**

最近、多くの Web サイトで暗号化された HTTPS 通信が利用されていますが、マルウェアによる C2 サーバとの通信においても HTTPS が使われることが多くなっています。そのため、暗号化された HTTPS のアウトバウンド通信を Web プロキシサーバで復号し、平文の HTTP と同様のログを記録することは、有効な対策になります。

HTTPS 通信は、明示型の構成の場合に CONNECT メソッドのみが記録されることは、先に紹介しましたが、Web プロキシサーバで復号していた場合、CONNECT メソッド以降の通信についてもログを取得することができます。

```
CONNECT outlook.office365.com:443 HTTP/1.1
POST https://outlook.office365.com/EWS/Exchange.asmx HTTP/1.1
```

**図 2-2 復号された HTTPS 通信のログの例**

また HTTPS 通信では、証明書を使用してサーバが正しい事の証明と暗号化通信を行います。証明書にはドメイン名が埋め込まれており、ブラウザは接続先のドメイン名と証明書のドメインが一致するか照合して、通信先のサーバが正しい事を確認しています。正規の Web サイトであれば、HTTPS の通信先として IP アドレスが使用されることはほとんどありません。そのため、IP アドレスでの HTTPS 通信を拒否しておくことで、マルウェアが IP アドレスで指定された通信先に HTTPS 通信することを防ぐことが可能です。 ただし、この対策は業務に影響が出る可能性があるため、次の③の注意点を合わせて確認ください。

また、HTTPS 通信の復号については、他にも注意すべき点があるため、後述の「HTTPS 通信の復号」も参照してください。

**③ 許可するポート番号は、HTTP/HTTPS それぞれの標準ポート 80/443（TCP）に限定する**

Web プロキシサーバからインターネット上の Web サーバに対して許可する通信は、HTTP で利用する TCP の 80 番および HTTPS で利用する TCP の 443 番ポートのみを許可します。これにより、マルウェアが標準外のポート番号を使用して通信を行おうとしても、通信を防ぐことが可能となります。

また、同様の理由で、Web プロキシサーバにおいても、CONNECT メソッドのリクエストは、TCP の 443 番ポートのみに制限しておきましょう。

ただし、CONNECT メソッドを利用した通信で、通信先を IP アドレスで指定したり、標準以外のポートを使用する正規のアプリケーションも存在しているため、上記の対策を実施すると業務に影響が出る可能性があります。この対策を実施する際には、事前に Web プロキシサーバのログを調査して、正規のアプリケーションが使用していないか確認した上で実施してください。

**④ 可能であればユーザ認証を導入する**

Web プロキシサーバで、ユーザ認証を導入するメリットは 2 つあります。1 つ目は、ユーザ認証機能に対応していない多くのマルウェアは認証をパスできず、C2 サーバとの通信に失敗する点です。2 つ目は、ユーザ認証機能に対応したマルウェアであっても、Web プロキシサーバのログには認証したユーザ名が記録されるため、ログから利用者を迅速に特定することが可能です。特に APT 攻撃で使われる遠隔操作型のマルウェアについては、盗んだ認証情報を利用する機能があるため、Web プロキシサーバでのユーザ認証に過信は禁物です。

なお、ユーザ認証を導入する際は、認証方式として Basic 認証は使用しないでください。Basic 認証は、入力されたユーザ・パスワード情報を暗号化せずに Web プロキシに送信するため、パケットキャプチャにより容易に認証情報が漏えいします。

**⑤ ログは最低でも 1 年間は保存する**

サイバー救急センターで対応した APT 攻撃のインシデントでは、長い場合には 1 年以上も被害に気がついていないことがあります。その場合、1 年以上前のログに遡って分析する必要があります。そのため、少なくともログは 1 年間は保存しておき、必要になったときにすぐに参照できるようにしておきます。

**⑥ ログの取得項目を確認する**

Web プロキシサーバで取得するログの項目は、機器のデフォルトの設定では不十分な場合があります。ログの取得項目をカスタマイズできる場合には、以下の項目を取得しておくとインシデント対応の際に有効です。

**表 2-1 Web プロキシサーバで取得すべき項目**

| No. | 項目 | 説明 |
|---|---|---|
| 1 | 時刻 | 通信が発生した時刻 |
| 2 | ユーザ名 | 認証されたユーザ名<br>（ユーザ認証していた場合） |
| 3 | 通信元 IP アドレス | 通信の送信元 IP アドレス |
| 4 | 応答コード | Web サーバからの応答コード<br>（例：403） |
| 5 | リクエストヘッダ | リクエストのヘッダで、メソッド、URL、パラメータ、HTTP バージョンが全て記録されること。特に、パラメータ（”?”以降の文字列）はデフォルトでは記録されないことがあるため、記録されていない場合には設定を確認する。<br>（例：POST /example/example.asp?param1=data1&param2=data2 HTTP/1.1） |

| | | |
|---|---|---|
| 6 | URL フィルタ結果 | URL フィルタのカテゴリ名や、結果など<br>（URL フィルタ等を使用していた場合） |
| 7 | メディアタイプ | リクエストに含まれるメディアタイプ<br>（例：text/html） |
| 8 | リクエストバイト数 | リクエストのサイズ。可能であれば、端末からのリクエストと、Web プロキシからのリクエストの両方を取得する |
| 9 | レスポンスバイト数 | レスポンスのサイズ。可能であれば、Web サーバからのレスポンスと、Web プロキシからのレスポンスの両方を取得する |
| 10 | ユーザエージェント名 | リクエストに含まれる User-Agent ヘッダ |
| 11 | 名前解決結果 | リクエストの宛先がドメインだった場合に、Web プロキシで名前解決を行った結果 |
| 12 | 通信先 IP アドレス | 通信の宛先 IP アドレス<br>多段構成の場合には上流の Web プロキシの IP アドレスになり、多段でない場合は上記の「名前解決結果」と同じになる |
| 13 | Referer ヘッダ | リクエストに含まれる Referer ヘッダ |
| 14 | X-Forwarded-For ヘッダ<br>※ | リクエストに含まれる X-Forwarded-For ヘッダ<br>※Web プロキシが多段になっており、手前の Web プロキシで該当ヘッダが付与される場合に設定する。“X-”で始まる異なるヘッダ名の場合もある。 |

**⑦ 不正な通信をいつでも遮断できるようにしておく**

マルウェアの感染に限らず、不審な通信が確認された場合には、該当の通信を Web プロキシサーバで遮断する必要があります。実際にサイバー救急センターで対応した過去のインシデントでは、以下のような対応を行ったことがあり、迅速に遮断できるよう準備しておくと、スムーズに対応可能です。

**【ケース１：全ての HTTP/HTTPS 通信を遮断する】**

APT 攻撃などの深刻なインシデントが発生しており、侵害されている端末や C2 サーバの通信先が特定できないような状況では、Web プロキシサーバやファイアウォールで HTTP/HTTPS 通信を全遮断することがあります。全遮断の状況でも、Web プロキシサーバがあれば、端末から発生した通信はログに記録されるため、通信遮断後に侵害されている端末を調査することが可能です。

**【ケース２：通信元の IP アドレスで遮断する】**

特定の端末からの全ての HTTP/HTTPS 通信を遮断します。この場合、他の端末から不審な通信が発生していた場合には、被害の拡大を止められないため、あわせて次の対応も行います。

**【ケース３ ：通信先のドメイン名や IP アドレスで遮断する】**

不審な通信として確認された通信先のドメイン名や IP アドレスを遮断することで、同様の通信先と通信している端末が他に存在していても、被害の拡大を防ぐことができます。

## (3) HTTPS 通信の復号

Web プロキシサーバで HTTPS 通信を復号すると、通常の HTTP 通信と同様に調査やフィルタリングが可能となるため多くのメリットがありますが、Web プロキシサーバの負荷は高まります。ネットワーク構成やパフォーマンスの検討は十分に行ってください。加えて、以下の点に注意しましょう。

### ① ルート証明書のインポート

経路上の Web プロキシサーバで HTTPS 通信を復号する場合、Web サーバから返された証明書とは異なる証明書をクライアントに返すことになります。クライアントでは、その証明書を受け入れるため、Web プロキシサーバで使用するルート証明書をあらかじめインポートしておく必要があります。一般的には OS に設定しておけば良いのですが、ブラウザなどのアプリケーションでも設定が必要な場合があります。

また、一部のアプリケーションは、ルート証明書のインポートに対応していなかったり、設定しても想定通りに機能しない場合があります。その場合には、Web プロキシサーバの設定で、該当のドメインに対して復号を行わないような例外設定で回避する必要があります。

### ② Web プロキシサーバの SSL/TLS 設定

Web プロキシサーバで HTTPS 通信の復号を行う際に、Web プロキシサーバの SSL/TLS の設定に問題があると、本来は接続を許可すべきではないサイトに対して、ユーザへの警告なしに接続が許可される可能性があります。そのため、接続先の Web サーバの証明書などに問題があった場合は、クライアントに警告を返すか接続を中止する設定にします。また、Web プロキシサーバが使用する SSL/TLS のバージョンが最新であり、脆弱な SSL バージョンを利用しない設定になっていることも確認します。これらの設定が有効であるかは、以下のサイトで確認できます。

https://badssl.com/dashboard/

### ③ 他のセキュリティ機器との連携

Web プロキシサーバで復号を行うことで、復号後の通信に対して URL フィルタやコンテンツチェックが可能になります。また、機器によっては、復号後の通信を他のセキュリティ製品と連携させることができるため、他のセキュリティ製品を使用している場合には検討するとよいでしょう。

# 脅威分析報告

## 「Mirai」ボットネットの今

### (1)「Mirai」ボットネットについて

2017 年 11 月頃から、「Mirai」ボットネットの感染が国内で急激に増えている事を確認しており、原因を調査した結果、「Mirai」または亜種の「Okiru」および「Satori」によるものであることを確認しました。そこで、今回はサイバー救急センターの脅威分析チーム(CTT: Counter CyberThreat Team)が確認した「Mirai」ボットネットに関する最新動向について紹介します。

始めに、「Mirai」ボットネットは 2016 年 10 月頃に開発者によってソースコードがリークされ、その後、セキュリティ研究者によってリークされたソースコードがオープンソースの Web サイトに公開されました。

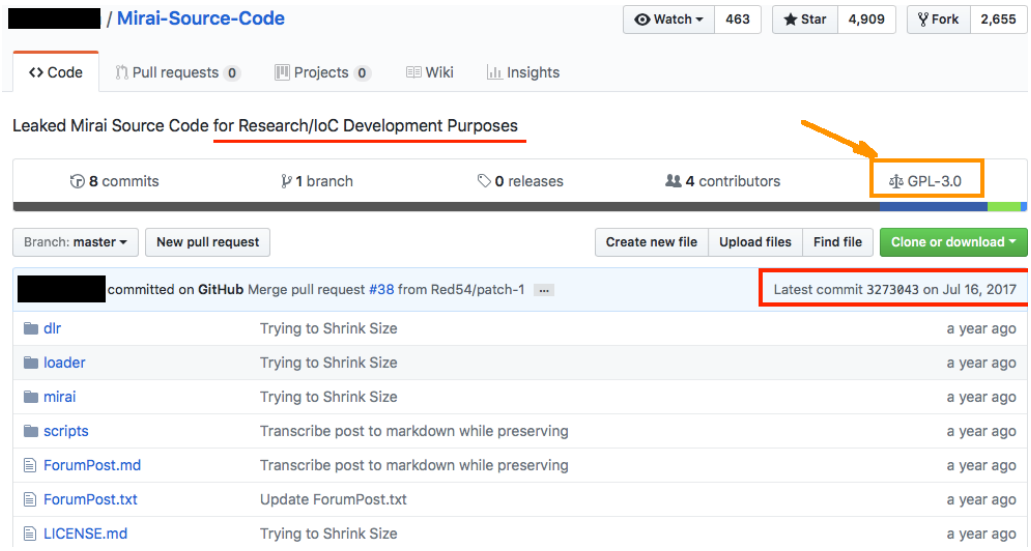


**図 3-1　オープンソースの Web サイトに公開された「Mirai」ボットネットのソースコード**

「Mirai」ボットネットのソースコードが公開された時点から、DDoS 攻撃に利用されていた Linux DDoS ボットネットが、「Mirai」ボットネットに入れ替わり始めました。CTT が調査した結果では、2017 年の 10 月の時点で Linux DDoS ボットネットの 82%が「Mirai」ボットネットになっており、「Mirai[1]」と、そ

1 リークされたソースコードから少しずつバージョンアップされています。

の亜種の「Okiru」および「Satori」の 3 種類のマルウェアが利用されています。

## (2)「Mirai」について

現在利用されている「Mirai」は、2016 年 10 月にリークされた従来の「Mirai」と比べ、以下に示す 3 つの機能が追加されています。

① バイナリの難読化とデータの暗号化
② 自動実行機能
③ IoT Watchdog の監視機能強化

### ① バイナリの難読化とデータの暗号化

図 3-2 は、従来の「Mirai」と現在の「Mirai」のバイナリデータを比較したものです。従来の「Mirai」のバイナリデータ内に確認できた文字列が、現在の「Mirai」では暗号化され可読性がほぼなくなっています。これは、ウイルス対策ソフトウェアによる検知を回避することが目的であると考えられます。

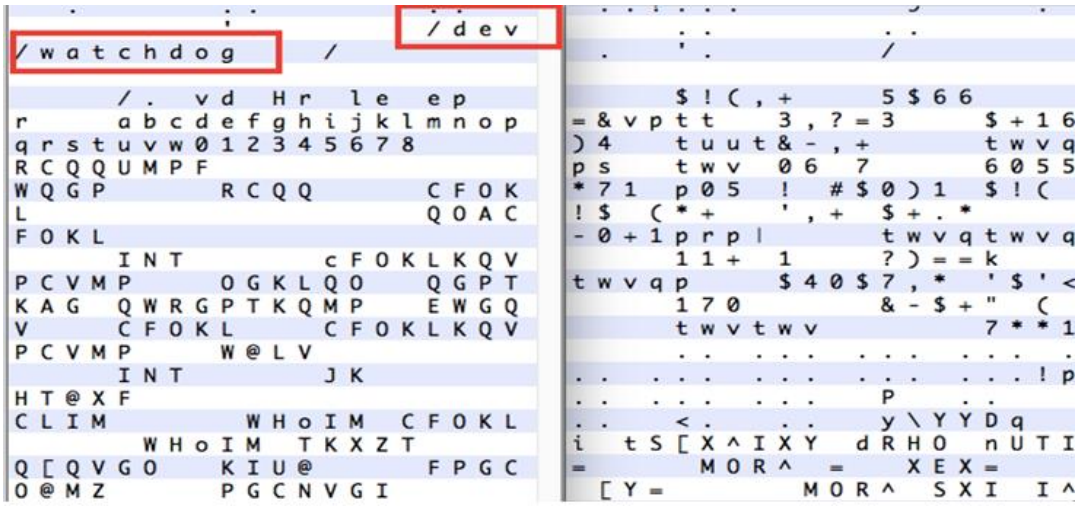


**図 3-2 「Mirai」のバイナリデータの比較：従来（左）／ 現在（右）**

### ② 自動実行機能

現在の「Mirai」には、/etc/rc.d/rc.local または/etc/rc.local などを利用して、自身を自動起動する永続性の仕組みが追加されています。これにより、「Mirai」に感染した IoT 機器を再起動したとしても、起動時に「Mirai」マルウェアが実行されて、感染状態が継続します。

これは、書き込み可能な領域を持った IoT 機器が対象となりますが、そのような IoT 機器が増えているという状況も背景にあると考えられます。

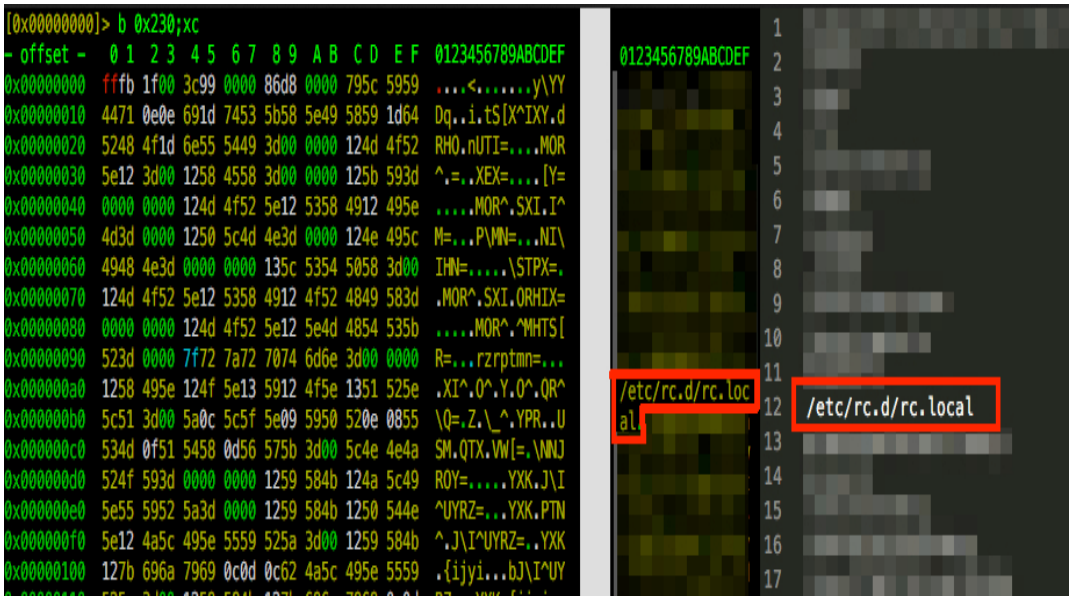


**図 3-3　現在の「Mirai」に含まれる自動実行の仕組み(/etc/rc.d/rc.local)：**
**暗号化された文字列（左）／ 復号した文字列（右）**

### ③ IoT Watchdog の監視機能強化

現在の「Mirai」は IP カメラの Watchdog 監視機能が強化されています。従来の「Mirai」にも Watchdog の監視機能のコード(図 3-4 の緑枠)がありましたが、Watchdog の設定ファイルに記載された停止機能や再起動機能のコード（図 3-4 の赤枠）が、現在の「Mirai」には追加されました。これは、「Mirai」自身の実行状態を維持するための改良と考えられます。


**図 3-4　IoT 機器の Watchdog 機能の悪用**

## (3)「Okiru」について

「Okiru」は「Mirai」の亜種であり、「Mirai」に比べ以下のような機能が追加されています。

① HUAWEI 製の HG532 ルーターの脆弱性(CVE-2017-17215)の悪用
② 辞書攻撃を行う際の認証情報の拡充
③ 独自の設定情報(コンフィグ)

### ① HUAWEI 製の HG532 ルーターの脆弱性(CVE-2017-17215)の悪用

HUAWEI 製の HG532 ルーターの脆弱性を悪用する 4 種類の攻撃コードが含まれています。脆弱性を悪用された場合、図 3-5 の黄枠で囲うコマンドがそれぞれ実行されます。


**図 3-5 「Okiru」に含まれる CVE-2017-17215 の脆弱性を悪用するコード**

### ② 辞書攻撃を行う際の認証情報の拡充

「Okiru」は、現在の「Mirai」に含まれていない Wifi IP カメラ、DVD レコーダー、Wifi ルーターなどの IoT 機器が工場出荷時に設定されている認証情報を保持しています。また、最新バージョンの「Okiru」では、CenturyLink や Qwest モデムなどの IoT 機器の認証情報が追加されました。これらの IoT 機器のパスワードを変更せずに使用していた利用者は意外に多く、アルゼンチンでは 2017 年 11 月末に大規模な「Okiru」の感染事例が発生しました。

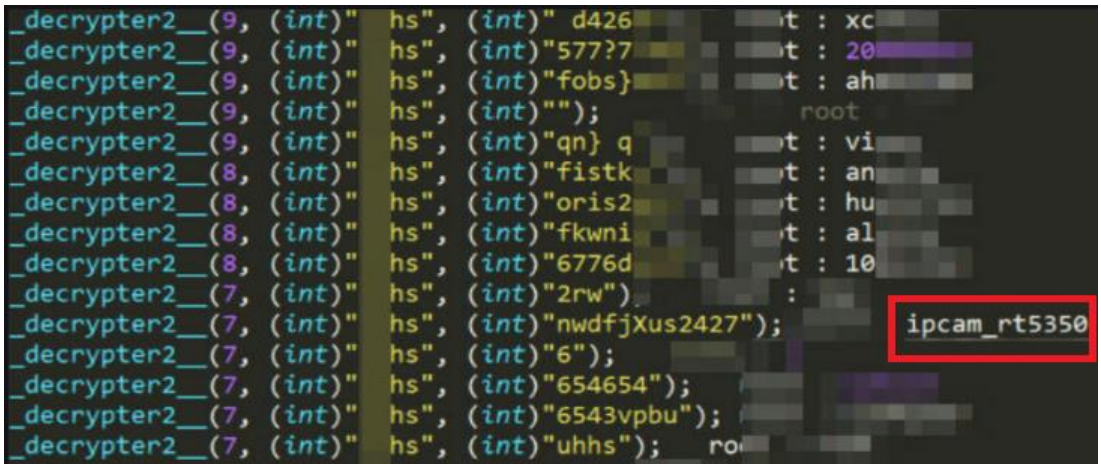


**図 3-6 「Okiru」に含まれる認証情報例**

### ③ 独自の設定情報(コンフィグ)

「Okiru」の設定情報は、「Mirai」とは異なり、２つのパートに分かれます。前半部分にはマルウェアの保存場所、ファイル名、プロセス名、コマンドなどが書かれており、後半部分には Telnet でやり取りする際の認証情報が書かれています。


**図 3-7 設定情報の一例**

## (4)「Satori」について

「Satori」も「Mirai」の亜種です。機能は、「Mirai」の機能を踏襲したものですが、先に紹介した「Okiru」とも異なっています。「Satori」の特徴は、主に以下の点になります。

① HUAWEI 製の HG532 ルーターの脆弱性(CVE-2017-17215)を悪用する攻撃コードを含まない
② 認証情報が「Okiru」に比べ少ない
③ 設定情報が「Mirai」と同様に分割されておらず、「Satori」という文字列が含まれる

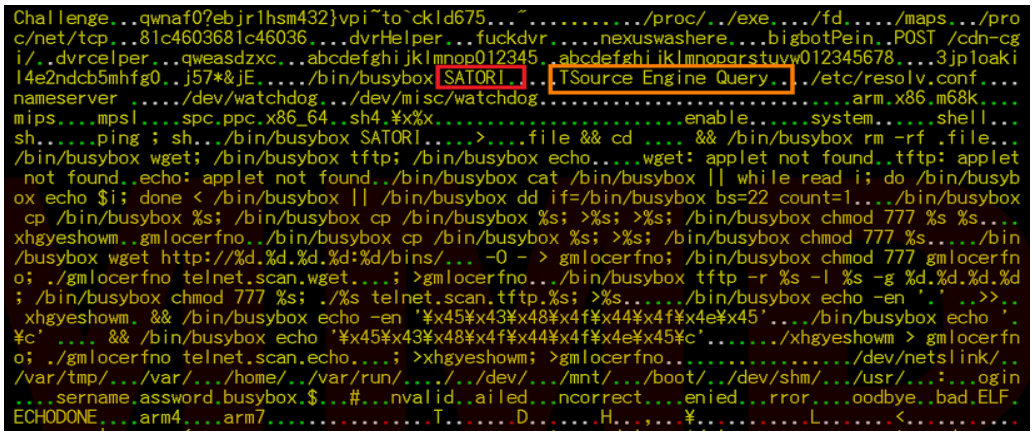


**図 3-8 「Satori」のコンフィグ**

## (5)「Mirai」、「Satori」、「Okiru」の比較

表 4-1 は、ここまで紹介した「Mirai」とその亜種の「Satori」および「Okiru」の主な機能を比較したものです。

**表 3-1 「Mirai」亜種の比較表**

| No | 機能 | 「Mirai」 | 「Satori」 | 「Okiru」 |
|---|---|---|---|---|
| 1 | 設定情報の分割 | なし | なし | あり |
| 2 | 自動実行機能 | なし[2] | あり | あり |

2 「Mirai」のバージョンによっては、自動実行機能を有している場合もある。

| | | | | |
|---|---|---|---|---|
| 3 | CVE-2017-17215 を悪用する攻撃コードの有無 | なし | なし | あり |
| 4 | 辞書攻撃するリストの多さ | 少ない | 多い<br>(図 3-9) | かなり多い<br>(図 3-9) |
| 5 | DDoS 機能(Reflection Attack) | あり | あり | なし |
| 6 | シェルコマンドの有無及びダウンロードコマンドの有無 (図 3-10) | あり | あり<br>(カスタム) | あり<br>(種類が豊富) |
| 8 | ダウンローダー機能<br>(図 3-11) | あり | あり<br>(カスタム) | あり<br>(カスタム) |
| 9 | Watchdog の対応機種の多さ | 少ない | 対応機種が多い | 対応機種が多い |
| 10 | ファイルの保存ディレクトリ名 | dvdHelper, | dvdHelper,<br>dvrRunNer,<br>dvrRapist,<br>dvrAssist,<br>dvrTelnet,dvr<br>Botnet | dvdHelper,<br>dvrRunNer,<br>dvrRapist,<br>dvrAssist,<br>dvrTelnet,<br>dvrBotnet |
| 12. | 対応 CPU | arm5,arm7<br>m68k,mips<br>mipsel,ppc,<br>sparc,sh4 | 「Mirai」<br>＋x86-64,<br>＋i686, i586<br>＋arm4,arm6 | 「Satori」<br>＋ ARC コア cpu<br>(図 3-12)[3] |


**図 3-9 「Okiru」(左)と「Satori」(右)の認証情報の比較**

3 2018 年 1 月初旬に ARC プロセッサに対応した「Okiru」の亜種を発見しました。

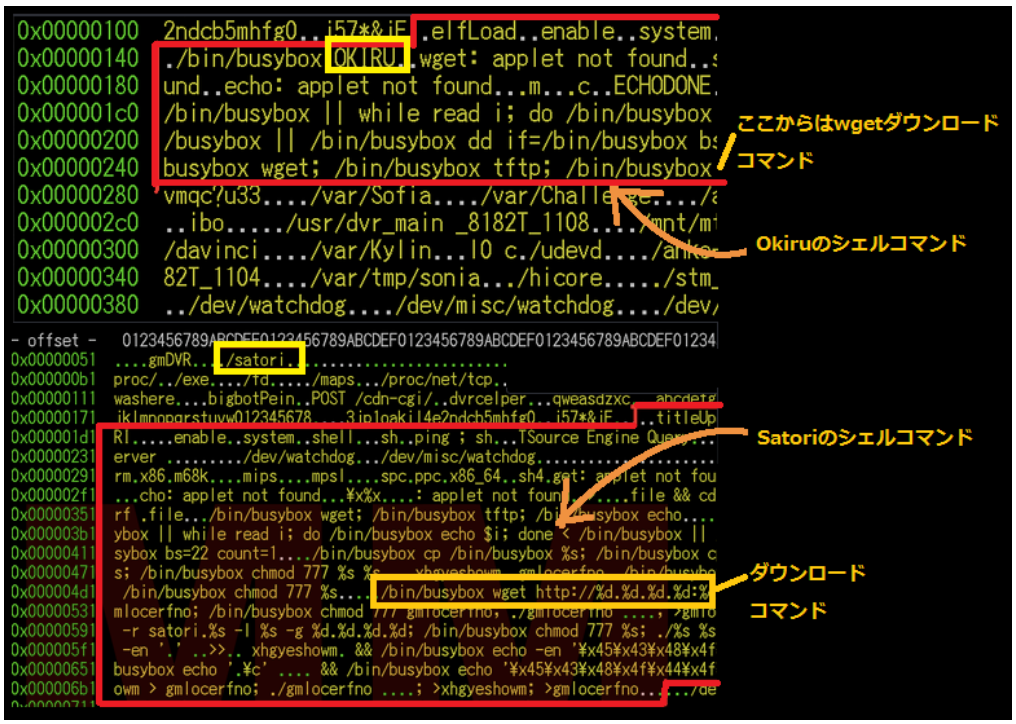


**図 3-10 「Okiru」(上)と「Satori」(下)のシェル及びダウンロードコマンドの比較**

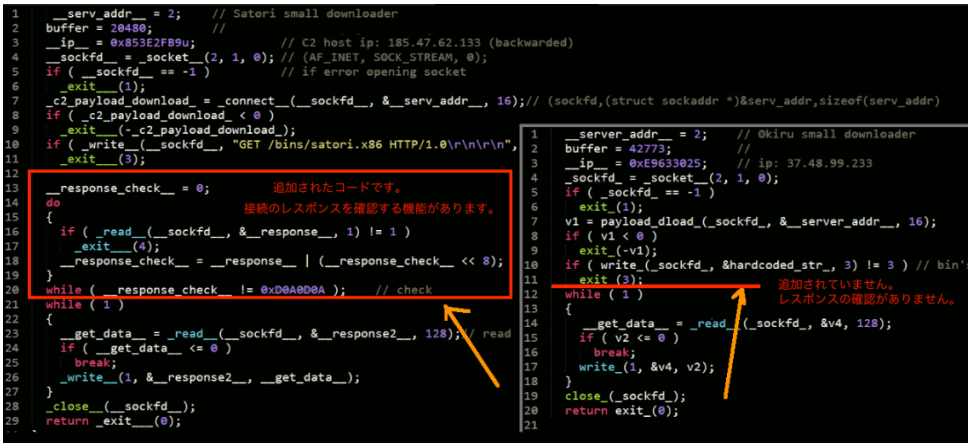


**図 3-11 「Satori」(左)と「Okiru」(右)のダウンローダー機能の比較**

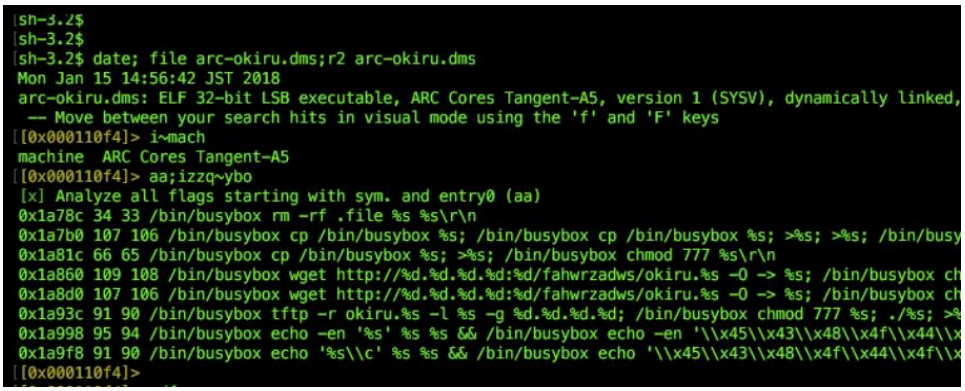


**図 3-12　ARC プロセッサに対応した「Okiru」の亜種**

## (6)「Mirai」感染を検知するために

最後に、Yara を利用して「Mirai」を見つける方法を紹介します。「Mirai」を検知するための Yara ルールは、「Mirai」が発見された 2016 年頃に作成されましたが、現在流行する「Mirai」の亜種は、ほとんど検知することができません。

このため、図 3-13 に示す方法を用いて yara ルールを作ることで、検知率を高めることができると考えています。以下に、IoC に加えて、「Okiru」と「Satori」に対応した Yara ルールを記載しますので、ご活用いただければ幸いです。

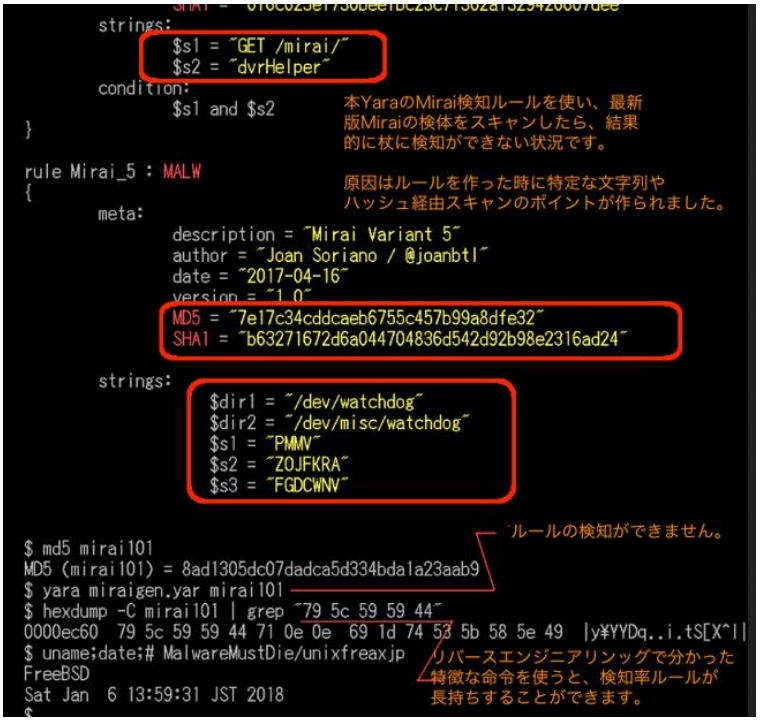


**図 3-13　「Mirai」マルウェアの Yara ルールを作る方法**

**【IOC 情報】**

MD5 (Satori-arm) = 977534e59c72dafd0160457d802f693d
MD5 (Satori-mips) = 27d6fb9b8af8408ca6ce2831762fa021
MD5 (Satori-arm) = cc2e611a511d4d907a6d39f552cc81df
MD5 (Satori-4mips) = a4abd90ea1a1a93e2b813abd380eda94
MD5 (Okiru-arc) = 9c677dd17279a43325556ec5662feba0
MD5 (Okiru-mips) = 214d8e84823bfba7adfe302aa6786d5a
MD5 (Okiru-x64) = 59e3fa23e0282368628b2dfe1605ccf7
MD5 (Okiru-x86) = fc11c9cb0d4433143271f0f767864a30
MD5 (Okiru-arm7) = c892092f58761b29dbd965b977412c10
MD5 (Satori-downloadr-arm) = ff06b2584f44e24b517074230c8de6e9
MD5 (Satori-downloadr-mips) = e8abfd033843b4504797eceaf825a118
MD5 (Satori-downloadr-x86) = 23502d13bf247c576ceaad28332cb121
MD5 (Okiru-downloadr1-mips) = 17bdf1e6692bba7ee19fc837a457d122
MD5 (Okiru-downloadr2-mips) = 24fc15a4672680d92af7edb2c3b2e957
MD5 (Okiru-downloadr3-ppc) = 5fb5d4a3f43a1202a973fc8328aede57
MD5 (Okiru-downloadr4-sh) = 808eaf4b336880a5d38a1d690fbd46b6
MD5 (Okiru-downloadr5-x8632) = 8d5b7fecf4894648394a60829cf0d29f
MD5 (Okiru-downloadr6-sparc) = 4215c48693e00cc683ed80bd3da10c3b
MD5 (Okiru-downloadr7-M68k) = 634b99b656cfefeafe4504c2ac1f9ddd
MD5 (Okiru-downloadr8-ARM) = 62112cf78affd879c8dcef2f3e62077f

**【Yara ルール】**

① Okiru.yar

```
/* Yara rule to detect Mirai Okiru generic
   This Yara ruleset is under the GNU-GPLv2 license (http://www.gnu.org/licenses/gpl-2.0.html)
   and   open to any user or organization, as long as you use it under this license.
*/

private rule is__Mirai_gen7 {
        meta:
                description = "Generic detection for MiraiX version 7"
                reference = "http://blog.malwaremustdie.org/2016/08/mmd-0056-2016-linuxmirai-just.html"
                author = "unixfreaxjp"
                org = "MalwareMustDie"
                date = "2018-01-05"

        strings:
        $st01 = "/bin/busybox rm" fullword nocase wide ascii
        $st02 = "/bin/busybox echo" fullword nocase wide ascii
        $st03 = "/bin/busybox wget" fullword nocase wide ascii
        $st04 = "/bin/busybox tftp" fullword nocase wide ascii
        $st05 = "/bin/busybox cp" fullword nocase wide ascii
        $st06 = "/bin/busybox chmod" fullword nocase wide ascii
                $st07 = "/bin/busybox cat" fullword nocase wide ascii

        condition:
        5 of them
}

private rule is__elf {
        meta:
                author = "@mmorenog,@yararules"

        strings:
                $header = { 7F 45 4C 46 }

        condition:
                $header at 0
}

rule Mirai_Okiru {
        meta:
                description = "Detects Mirai Okiru MALW"
                reference =
"https://www.reddit.com/r/LinuxMalware/comments/7p00i3/quick_notes_for_okiru_satori_variant_of_mirai/"
                date = "2018-01-05"

        strings:
                $hexsts01 = { 68 7f 27 70 60 62 73 3c 27 28 65 6e 69 28 65 72 }
                $hexsts02 = { 74 7e 65 68 7f 27 73 61 73 77 3c 27 28 65 6e 69 }
                // noted some Okiru variant doesnt have below function, uncomment to seek specific x86 bins
      // $st07 = "iptables -F¥n" fullword nocase wide ascii

        condition:
                all of them
                and is__elf
                and is__Mirai_gen7
                and filesize < 100KB
}
```

② Satori.yar

```
/* Yara rule to detect Mirai Satori generic
    This Yara ruleset is under the GNU-GPLv2 license (http://www.gnu.org/licenses/gpl-2.0.html)
    and   open to any user or organization, as long as you use it under this license.
*/
private rule is__Mirai_gen7 {
                meta:
                                description = "Generic detection for MiraiX version 7"
                                reference = "http://blog.malwaremustdie.org/2016/08/mmd-0056-2016-linuxmirai-just.html"
                                author = "unixfreaxjp"
                                org = "MalwareMustDie"
                                date = "2018-01-05"

                strings:
                $st01 = "/bin/busybox rm" fullword nocase wide ascii
                $st02 = "/bin/busybox echo" fullword nocase wide ascii
                $st03 = "/bin/busybox wget" fullword nocase wide ascii
                $st04 = "/bin/busybox tftp" fullword nocase wide ascii
                $st05 = "/bin/busybox cp" fullword nocase wide ascii
                $st06 = "/bin/busybox chmod" fullword nocase wide ascii
                $st07 = "/bin/busybox cat" fullword nocase wide ascii

                condition:
                5 of them
}
private rule is__elf {
                meta:
                                author = "@mmorenog,@yararules"
                strings:
                                $header = { 7F 45 4C 46 }

                condition:
                                $header at 0
}

private rule is__Mirai_Satori_gen {
                meta:
                                description = "Detects Mirai Satori_gen"
                                reference =
"https://www.reddit.com/r/LinuxMalware/comments/7p00i3/quick_notes_for_okiru_satori_variant_of_mirai/"
                                date = "2018-01-05"

                strings:
                                $st08 = "tftp -r satori" fullword nocase wide ascii
                                $st09 = "/bins/satori" fullword nocase wide ascii
                                $st10 = "satori" fullword nocase wide ascii
                                $st11 = "SATORI" fullword nocase wide ascii

                condition:
                                2 of them
}
rule Mirai_Satori {
                meta:
                                description = "Detects Mirai Satori MALW"
                                date = "2018-01-09"

                strings:
                                $hexsts01 = { 63 71 75 ?? 62 6B 77 62 75 }
                                $hexsts02 = { 53 54 68 72 75 64 62 }
                                $hexsts03 = { 28 63 62 71 28 70 66 73 64 6F 63 68 60 }

                condition:
                                all of them
                                and is__elf
                                and is__Mirai_gen7
                                and is__Mirai_Satori_gen
                                and filesize < 100KB
}
```

# コラム：セキュリティ百景 #3

## 京都府警察研修

2017 年 9 月 26 日～27 日、京都府警察学校にてサイバー特別捜査官研修生に対する研修を実施しました。

多くの都道府県では、IT 系民間企業の経験者をサイバー特別捜査官として中途採用している中、京都府警は中途採用を行わずに基礎から育成するシステムを 2014 年にスタートさせました。このシステムが目指すものは、警察庁技官などの技術者と IT に不慣れな警察官の間の技術面における橋渡し人材の育成です。

選抜者を対象にサイバー犯罪対策課などでの実地研修を行いながら、民間企業などからの派遣講師による定期的な研修を繰り返し受講させ、研修修了者をサイバー特別捜査官に任命します。

弊社も発足時から本育成の支援をさせていただいており、本年は「デジタルフォレンジックに関する講義」、「新しい捜査手法を考えるグループディスカッション」を実施しました。

**写真 4-1　講義風景**

両講義とも好評でしたが、中でも「新しい捜査手法を考えるグループディスカッションは今までにない斬新な内容で他の捜査員にもぜひ受講させたい」との感想を多くいただきました。

この講義では、私が以前県警のサイバー特別捜査官として働いていた際の事例や経験、常に考えていたことをメッセージとしてお伝えしました。

今後も機会があれば全国警察のサイバー犯罪捜査能力の向上にお手伝いしていきたいと考えています。

**写真 4-2　警察学校近くのこってりラーメン**

おわりに、皆さんにどうしてもお伝えしたいことを。

京都には何度も訪れていましたが、今回はじめて気付いたことがあります。

それは「ラーメン屋」の多さです。聞くところでは、私の京都イメージとは正反対のこってり味のお店が多いとか。学校近くのお店もとても美味でした。

皆さんも京都を訪れた際は、一食こってりラーメンを食べてみることをおススメします。

サイバー救急センター：佐藤 敦

# コラム：セキュリティ百景 #4

## 三重県警察研修

2017 年 10 月 30 日(月)～11 月 2 日(木)に、三重県警察にて「サイバー空間への対処に関する共同対処協定締結事業者社員との情報交換プログラム」が行われ、弊社からは私を含めた 3 名が参加しました。

三重県警察と弊社は 2016 年に「サイバー空間の脅威に対する共同対処協定」を結んでおり、人材交流についても協定の一環として実施しています。2016 年に 1 回目が実施され、今回は 2 回目の実施になります。

**写真 5-1　模擬演習の状況**

内容については、三重県警察側からは警察組織の説明や施設の案内をしていただいたり、サイバー犯罪捜査の模擬演習を体験させていただきました。

弊社側からは弊社研究部門のサイバー・グリッド・ジャパンに蓄積されたセキュリティ技術者養成のための知見を活用した CTF 運営のノウハウや、三重県内での産学官連携などについて、意見交換を行いました。

今回の交流を通じて、証拠を集める難しさを実感しました。また、調査を行っているとつい忘れがちになってしまいますが、攻撃者は人間であるということを、今回の模擬演習で再認識することができました。

最後に、私は今回初めて三重を訪れたのですが、関東と違い皮をパリパリに焼いて食べるのが三重のうなぎで、とてもおいしかったです。三重を訪れた際は、ぜひ御賞味ください。

**写真 5-2　三重の名物「津のうなぎ」**

サイバー救急センター：髙橋　勇介

# 編集後記

無事に 2 号をお届けすることができて、ホッとしております。ご協力いただいた皆様に感謝いたします。最近、金銭目的の攻撃者グループが、現実通貨だけでなく仮想通貨も視野に入れてサイバー攻撃を行っております。そのような攻撃者グループは、サーバに不正侵入しても情報の窃取等は行わず、コインマイナーを動かして仮想通貨を発掘するための演算能力（≒電力）の窃取を目的としているため、重要な情報を保持していなくても狙われることになります。はじめにでも書きましたが、想定の内と外を改めて見渡し、セキュリティ対策を見直してみましょう。（法）

最後になりますが、本号でもアンケートへのご協力をお願いいたします。

## アンケートのお願い

今後のよりよい記事づくりの参考とさせていただくため、読者アンケートを実施いたします。以下の URL または QR コードから、忌憚のないご意見・ご感想をお寄せください。

https://jp.surveymonkey.com/r/CXNYBP8

編集長　内田 法道

編集者・執筆者　伊原 秀明、田原 祐介、関 宏介、高松 啓、石川 芳浩、アドリアン ヘンドリック、佐藤 敦、髙橋 勇介


**株式会社ラック**

〒102-0093 東京都千代田区平河町 2-16-1 平河町森タワー

TEL: 03-6757-0113 （営業）

E-MAIL: sales@lac.co.jp

https://www.lac.co.jp/


**緊急対応窓口：サイバー救急センター**

24時間365日 緊急対応コール

**0120-362-119**

CYBER EMERGENCY CENTER

ご相談は予約不要、24時間対応。すぐにご連絡ください。