



Reverse Engineering : Script Kiddie

Value : 100 pts

Difficulty : Unknown

Description : It looks like a script kiddie was trying to build a crypto locker. See if you can get the database back ?

Attachment : encrypted_db

Solution :

Download the file «encrypted_db» and open it.

6579776964584e6c636a45694f6e7369626d46745a534936496b746c5957
6468626942495a584a36623263694c434a3163325679626d46745a534936
496c4a7659326c7658307868596d466b0a61575531496977695a57316861
5777694f694a4b62334e6f64574666516d56705a58493051486c68614739
764c6d4e7662534973496d466b5a484a6c63334d694f6e7369633352795a
5756300a496a6f69516d396e61584e705932676754576c7a63326c766269
4973496e4e316158526c496a6f69515842304c6941344e4441694c434a6a
61585235496a6f69536d4635626d566962334a760a6457646f4969776965
6d6c775932396b5a534936496a63784e6a4d33496977695a325676496a70

It seem to be hexadecimal, all hash count number from 0 to 9 and letter from a to f. Let's decrypt all the hash from hexadecimal to text.

Source : <http://www.unit-conversion.info/texttools/hexadecimal/>

hex numbers to text

The decoded output seem to be a base64, let's decode it too.

Source : <https://www.base64decode.org/>

For encoded binaries (like images, documents, etc.) use the file upload form a bit further down on this page.

UTF-8

Source character set.

Live mode OFF

Decodes in real-time when you type or paste (supports only UTF-8 character set).

< DECODE >

Decodes your data into the textarea below.

```
{
  "user1": {
    "name": "Keagan Herzog",
    "username": "Rocio_Labadie5",
    "email": "Joshua_Beier4@yahoo.com",
    "address": {
      "street": "Bogisch Mission",
      "suite": "Apt. 840",
      "city": "Jayneborough",
      "zipcode": "71637",
      "geo": {
        "lat": "33.1223",
        "lng": "64.2738"
      },
      "phone": "017-397-1881",
      "website": "kyra.info",
      "company": {
        "name": "Labadie, Reinger and McGlynn",
        "catchPhrase": "User-centric demand-driven hardware",
        "bs": "compelling repurpose infrastructures"
      }
    },
    "user2": {
      "name": "Ashleigh Osinski",
      "username": "Hortense_Effertz15",
      "email": "Liliana.Feeney@gmail.com",
      "address": {
        "street": "Krajcik Gateway",
        "suite": "Suite 142",
        "city": "Carolchester",
        "zipcode": "38963-0862",
        "geo": {
          "lat": "37.3141",
          "lng": "-102.8027"
        },
        "phone": "(230) 125-5903",
        "website": "devin.net",
        "company": {
          "name": "Ankunding - Collins",
          "catchPhrase": "Organized regional challenge",
          "bs": "frictionless whiteboard vortals"
        }
      },
      "user3": {
        "name": "Iva D'Amore",
        "username": "Howell_Pfannerstill",
        "email": "Jarred_Deckow@gmail.com",
        "address": {
          "street": "Olson Glens",
          "suite": "Apt. 870",
          "city": "South Donaldburgh",
          "zipcode": "07601-7976",
          "geo": {
            "lat": "36.9448",
            "lng": "-96.7118"
          },
          "phone": "454.213.5840 x5846",
          "website": "petra.com",
          "company": {
            "name": "Gulgowski Inc",
            "catchPhrase": "Reactive bi-directional process improvement",
            "bs": "cross-platform orchestrate niches"
          }
        },
        "user4": {
          "name": "Augustine Kreiger",
          "username": "Kamron_Williamson89",
          "email": "Trevor_Hickle@gmail.com",
          "address": {
            "street": "Alisa View",
            "suite": "Apt. 797",
            "city": "Rosalynbury",
            "zipcode": "95522",
            "geo": {
              "lat": "77.4075",
              "lng": "122.8027"
            },
            "phone": "(230) 125-5903",
            "website": "devin.net",
            "company": {
              "name": "Ankunding - Collins",
              "catchPhrase": "Organized regional challenge",
              "bs": "frictionless whiteboard vortals"
            }
          }
        }
      }
    }
  }
}
```

Save the output into a file (i called it decrypted_db). Open it with leafpad or something and use the function «Search» for search the «flag» keyword.

```
"flag{ENC0D1NG_D4TA_1S_N0T_ENCRY7I0N}",
```

Flag : flag{ENC0D1NG_D4TA_1S_N0T_ENCRY7I0N}