|  |
| --- |
| CyberVets |

|  |
| --- |
| Cybersecurity Documentation Report |

## **Encryption Challenge**

## Custom Encryption

[Description: Can you get sense of this code file and write the function that will decode the given encrypted file content. Find the encrypted file here flag info and code file might be good to analyze and get the flag. TIP: Understanding encryption algorithm to come up with decryption algorithm.]

## Problem

I will need to decrypt the file and I gathered the following information from GtkHash on Linux from the “enc\_flag(3) file.

MD5: c8f28713418cbbd1921aae5d58af0082

SHA1: 1478482d56a7a99bc52a87500dcd6bcf18dd2c47

SHA256: 945e47d9090da9e88e3c77bced6a6ff004d666129e278781a42174d10316bdb5

CRC32: 851aa46b

Decryption Key found on <https://planetcalc.com>

Decoder Site: <https://www.dcode.fr/caesar-cipher>

Decryption Key: LPZDGTVBNKEJAHRSFMQIOUXYWC

Key used to Encrypt: MHZDKQENTLJARIUBSOPFVGYWXC

## Solution

1. Installed cryptography with “sudo apt install python3-pip”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Reverse Executable of a File (Reduce Size of Binary)**

Packer Challenge

[Description: Reverse the linux executable? I will need to use Reverse Engineering to reduce the size of the binary file after compiling it.]

Problem:

The upx commands that were discovered on the link in the “out” file are not producing the flag. I’m able to pack the file, unpack it and compress it, but no flags have been produced. Below is some more information that I have gathered from running certain commands.

I downloaded the file to my linux machine and was able to inspect it and retrieve the Checksums MD5, SHA1, CRC32 and SHA2256. After that I opened terminal from the file and ran “upx” to unpack the binary file “out” and ran -d command to compress it. I researched different syntax to reduce the size of the binary and received multiple basupxuph: syntax errors. I then ran a web program called “ASPOSE” to compress files and it failed to work on the file I was working on.

Ran upx -L file “out”

Ultimate Packer for eXecutables

Copyright (C) 1996 - 2024

UPX 4.2.2 Markus Oberhumer, Laszlo Molnar & John Reiser Jan 3rd 2024

This program may be used freely, and you are welcome to

redistribute it under certain conditions.

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

UPX License Agreements for more details.

You should have received a copy of the UPX License Agreements

along with this program; see the files COPYING and LICENSE.

If not, visit one of the following pages:

https://upx.github.io

https://www.oberhumer.com/opensource/upx/

Markus F.X.J. Oberhumer Laszlo Molnar

<markus@oberhumer.com> [ezerotven+github@gmail.com](mailto:ezerotven+github@gmail.com)

┌──(kali㉿DESKTOP-FU7LFEB)-[~/Desktop/PICO]

└─$ upx -d out

Ultimate Packer for eXecutables

Copyright (C) 1996 - 2024

UPX 4.2.2 Markus Oberhumer, Laszlo Molnar & John Reiser Jan 3rd 2024

File size Ratio Format Name

-------------------- ------ ----------- -----------

877724 <- 383324 43.67% linux/amd64 out

Solution:

1. So far none.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**The Basics of Wireshark**

Analyze Protocols and PCAPS

[Wireshark is an open-source, cross-platform network packet analyzer tool capable of sniffing and investigating live traffic and inspecting packet captures (PCAP). It is commonly used as one of the best packet analysis tools.]

This overview and challenge with Wireshark demonstrated to me the capabilities of sniffing and investigating live traffic and inspecting PCAPs. I learned how to dissect packets, navigate through the tool and understand the basics of packet filtering. There are two types of filtering approaches which are "Capture Filters" and Display Filters". Applying these filters is important to understand so the operator can effectively investigate traffic captures.

**Information CTF**

Forensics

[Description: Files can always be changed in a secret way. The goal of this CTF is to modify the data in the cat.jpg file and uncover hidden information.]

Steps:

1. I downloaded the file and ran “file cat.jpg” to verify it was a jpg image.

Problems:

1. I inspected the file but could not find the license information from the properties.
2. I could not download image viewers to my Kali Linux system that were free such as Gwenview on KDE Manjaro, Photoshop or GIM.

Solution:

1. I was able to locate the license for the jpg image and ran a “base64 -d” command.
2. I then input the license number and hit “ctrl+d” which displayed the hidden content in the license string.