Codefest Capture the flag Editorial

Contest Duration : Feb 26 2016, 12:00 am IST to Feb 27 2016, 11:59 pm IST

-> Problem 1: Web Basics

The flag was hidden on one of the link on the website. After successful authentication the page would redirect to this particular link. The authentication was being done using JavaScript on the same page. So basically we just had to view source on this one, bypass authentication, view the link and get the flag.

Flag: 2e746ccb4882d3a7ae21

-> Problem 2: Launch Codes:

Used 7zip to extract the given file, there was a file key.txt which contains the key.

Flag :3f5a57462f77faa850439989bf773df9

-> Problem 3: Login Details

Open file in hex editor it has the signature of normal image files, so change the extension from txt to jpg, png, bmp, etc. and get the username and password.

Flag: paint:rules

-> Problem 4: Looking back in Life!

The answer was '::1' the Ipv6 loopback address. Ipv6 strips leading zeroes and loopback is the address for testing network flow.

Flag: ::1

-> Problem 5: Guess Me

Check the properties of image in it's details section you will find the comments property with value as "I like chocolate".Since he likes chocolate this must be his password.

Flag: chocolate

-> Problem 6: BrightEdge Needs Help

There's a RAR File containing "key.txt" embedded at the end of the image.
Flag: It_WaS_CF'16

-> Problem 7: Weak Authentication

The site was using cookie to provide access. Just chang it's value to "True" to gain access.
Flag: d55f03db07878b51b

-> Problem 8: Ghost Recon

Open the pcap in wireshark , there was a lot of tcp stream and there get the flag in one of tcp stream in plain text .
Flag: vim_is_the_best

-> Problem 9: Whiteout

In this one the image was complete white, but on viewing the laptop screen on some another angle, some text was visible.
The text was "key{forensics_is_fun}". You can confirmed this by changing contrast of image in GIMP. Or Use the Stagsolve Tool.
Flag: forensicsis_fun

-> Problem 10: USB Modem

The given pcap file had usb traffic from a keyboard. You just had to decode each frame to get the flag.
Flag: c48ba993d353ca

-> Problem 11: Cryptograph

The given message was ascii encoded value in all three formats(oct, bin, dec). You just had to reverse it to get the key.
"Codefest invites all the hackers to show off their skills. here's your key{15e474d2946628dcd}. "
Flag: 15e474d2946628dcd

-> Problem 12: Indentation Matters

Copy the source code, will show nothing. When analysed in Sublime text editor shows some Dot and Dash got to know that it is Whitespace language.Analys code in disassembler, understand the assembly language code and get the password, entering the password will get you the key.

-> Problem 13: Easy RSA

Use openssl rsa utilities to decrypt the given file as:
Openssl asn1parse –I – in key.txt
In this one, RSA public key had very large modulus(1022 bit instead of 1024) as well as equally large (1021 bit) exponent. Such large keys are generally prone to Wiener's attack. So after performing wiener attack using scripts from a ctf-writeup , get the private exponent. Decrypted the bin file using openssl's rsa-utilities to get the key.
Flag: itbhu codefest ctf

-> Problem 14: looks can be deceiving

1st image is the GNU logo .
2nd image is Dennis Ritchie, the inventor of the C language and Unix .
3rd image is the Banaras Hindu University.

The most-often software used in steganography is Steghide, Data was hidden in justanormalimage.jpg. Get the flag using password "gnuunixbhu" (gnu + unix + bhu).
Flag: gnuunixbhu

-> Problem 15: crack me

After opening the binary in IDA, track the "correct" string
and you will find that a function (0x8048580) was being executed which was a switch-case construct. After a bit more inspection using decompiled C code , you can find that from each character of provided input, 10 was being subtracted and it's location and value was being checked . Using decompiled code, find most of the characters, the remaining can be found using IDA.
Flag: 09vdf7wefijbkh

-> Problem 16: Back to the Future 1

The language was the ancient language of star wars: "aurebesh". You can use this (http://www.aurebesh.swtor-tools.com/) awesome tool to get the message: "codefestresurrects".
But the key was in Camel Case : CodefestResurrects.
Flag: CodefestResurrects

-> 17: Fight Club

The video file had binary data, which was ascii encoding of a famous

dialog of Fight Club. The key was at the end of dialog:
{the things you used to own, now they own you} without brackets.
Flag: the things you used to own, now they own you

-> Problem 18: Old School

The provided string was flag encrypted as follows:

* first convert character to ascii values

* write the element's symbol corresponding to atomic number in place of ascii values

eg. Modified in elements from the periodic table:
Ho Au Er Tm Yb Md Bi Po Ho Tb Hg Po At Fl Tm Lv Rf Tm Yb Os Bk Lu In Xe
Using each element symbol to its atomic number:
67 79 68 69 70 101 83 84 67 65 80 84 85 114 69 116 104 69 70 76 97 71 49 54
Converted from decimal to ASCII:
Flag: CODEFeSTCAPTUrEthEFLaG16

-> Problem 19: Don't trust the user submitted files

After analysing packets in fiddler, yu will find that uploading any file with .gif extension results in mp4 file, and uploading non .gif extension results in error message. The library used for this process is FFmpeg.
The script used to convert the file should be something like:
ffmpeg -i file.gif file.mp4
The extension of the input file isn't important as its content will be parsed.
The idea is to provide a playlist, recognized by ffmpeg, which will open the /home/temp/key.txt file.
Thanks to the m3u file format it's quite easy, this is the modified gif:
#EXTM3U
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-TARGETDURATION:1
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0
#EXTINF:10.0,
file:///home/temp/key.txt
#EXT-X-ENDLIST
The content of the file will be output into the video stream.
Flag: 2e81333e94ec763

-> Problem 20: Key Lost in the Wires

This challenge provides us a file named poir. It is a pcap-ng file. Wireshark shows us more than 10k packages, mostly HTTP traffic. These packages are from the transfer of a file named “key.7z”. But this is not only one transfer. This file was send in 500 different pieces, which overlaps and in wrong order.

You can use the “Export Objects”-Option from Wireshark to save all 500 pieces. For the next step you have to think about how I you can get all files in the right order. You could extract the “Range: bytes=”- strings of the poir-file and save those to a new file. If you do this, you can sort the key.7z pieces based on the order of the range strings.A little python-scrip will do this task.

Now you can extract the files of this archive. There is only one file named “key.png”. It is a completely white picture. If you take a close look and see that not all pixels are completely white. Increase the visibility of these pixels.The result shows a png with the key.

Falg : Then I tried Pioneer_Of_Invisible_Road_By_BuGeun