CS 490 - Ethical Hacking
PicoCTF Report
Thomas Reilly

Binary Instrumentation

Binary instrumentation, medium

I've been learning more Windows API functions to do my bidding. Hmm…. I Swear this program was supposed to create a file and write the flag directly to the file. Can you try and intercept the file writing function to see what went wrong?

Hints:
Hint 1 - Frida is an easy-to-install, lightweight binary instrumentation toolkit
Hint 2 - Try using the CLI tools like frida-trace
Hint 3 - You can specify the exact function name you want to trace

Overview - binary instrumentation problem. Goal is to analyze or manipulate behavior of binary executables by inbedding outside code. All this change must be down at runtime without changing the original program. Intercept function calls and behavior analysis would be primary methods of approach.

Fridge - was tool kit recommended by the picoCTF hints. It was a toolkit for dynamic instrumentation. This allows JavaScript injections into running processes

Step 1: Download and extract the provided zip file
Download an exe file and unzip
Use provided password: picoctf

Wget
https://challenge-files-picoctf.net/c_varbal_sleep/4aee1b9778a8e56724d015b027431fb236853a94f53e5dcf32caed404da/bininst2.zip

Step 2: Run program
-launch file use password picoctf
-create file to write flag into it

Step 3: Use Frida toolkit to intercept file write
Run command: frida-trace -i "WriteFile" _bininst2.exe.extracted
Analyze extract file file 6000.7z is produced
Locate file using ls 6000.7z
Run file using file 6000
A String Base64 is discovered

Decode the string to text
Output should read: picoCTF{frida_f0r_bin_in5trum3nt4tion!_b21aef39}

BitLocker

Forensics, Medium
*problem could not be solved in webshell

Description: Jacky has learnt about the importance of strong passwords and made sure to encrypt the BitLocker drive with a very long and complex password. We managed to capture the RAM while this drive was opened however. See if you can break through the encryption.
Hint: Try using a volatility plugin

Overview: This challenge is forensics challenge using Bitlocker-encrypted environment. The goal will be to analyze and investigate data from this encrypted environment. Bitlocker has disk encryption built into it's platform. Goal will be to use python scripts to extract data

Step 1: download the disk image and RAM dump

Step 2: Search for plugin for volatility such as(*https://github.com/breppo/Volatility-BitLocker* .)

Step 3: Install python2 and pip2 for script development.

Step 4: Dump potential found FVEKs into a path defined --bitlockerFlag.

Step 5:Create mount points and unlock the bitlocker image using FVEK. Specify the path of the FVEK obtained --fvek flag

Step 6: Mount decryption Partition
Use Command: sudo mount -t ntfs-3g /mnt/bitlocker/bitlocker-file /mnt/bitlocker_decrypted

Step 7: use the mount point to get flag
Solution: picoCTF{B1tlock3r_driv3_d3crypt3d_9029ae5b}

FANTASY CTF

Easy, General Skills, picoCTF 2025, browser_webshell_solvable

Play this short game to get familiar with terminal applications and some of the most important rules in scope for picoCTF.
Connect to the program with netcat:
`$ nc verbal-sleep.picoctf.net 58889`

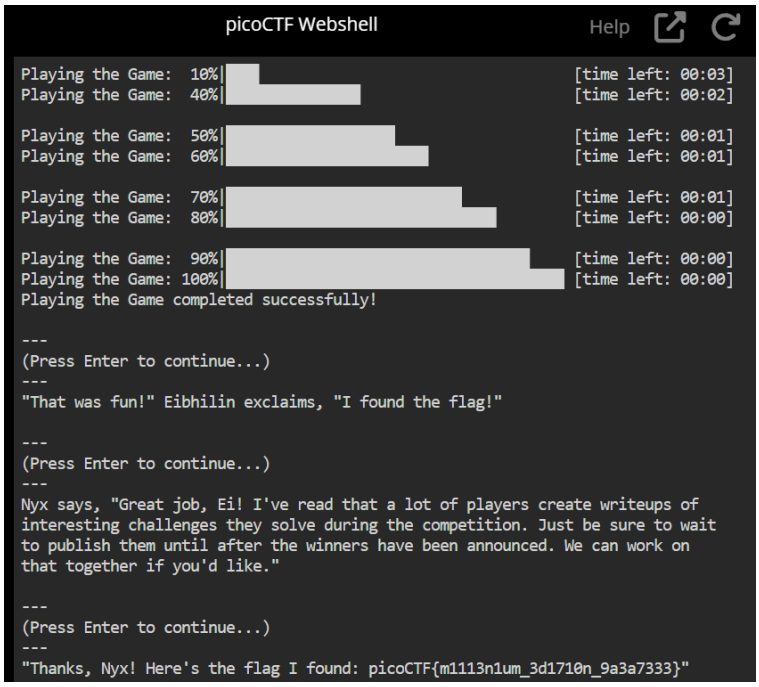
Hint: When a choice is presented like [a/b/c], choose one, for example: `c` and then press Enter.

Solution:
This challenge is terminal based, relying on terminal and Netcat commands

Step 1: connect to Netcat
nc verbal-sleep.picoctf.net 58889

Step 2: Work with input output prompts

Step 3: Using terminal explore program for flag
FLAG: picoCTF{m1113n1um_3d1710n_9a3a7333}



Verity

Easy, Forensics, grep, browser_webshell_solvable, checksum

People keep trying to trick my players with imitation flags. I want to make sure they get the real thing! I'm going to provide the SHA-256 hash and a decrypt script to help you know that my flags are legitimate. Additional details will be available after launching your challenge instance.

`ssh -p 50443 ctf-player@rhea.picoctf.net`

Using the password `6dd28e9b`. Accept the fingerprint with `yes`, and `ls` once connected to begin. Remember, in a shell, passwords are hidden!

- Checksum: 03b52eabed517324828b9e09cbbf8a7b0911f348f76cf989ba6d51acede6d5d8
- To decrypt the file once you've verified the hash, run `./decrypt.sh files/<file>`.

Hints:

Hint1 - Checksums let you tell if a file is complete and from the original distributor. If the hash doesn't match, it's a different file.

Hint2 - Checksums let you tell if a file is complete and from the original distributor. If the hash doesn't match, it's a different file.

Hint3 - Remember you can pipe the output of one command to another with `|`. Try practicing with the 'First Grep' challenge if you're stuck!

Solution:

SHA-256 hash must be located and decrypted and verified.
SHA-256 checksum hash: 03b52eabed517324828b9e09cbbf8a7b0911f348f76cf989ba6d51acede6d5d8
Use a decryption script like ./decrypt.sh

Step 1: connect to the SSH: `ssh -p 50443` ctf-player@rhea.picoctf.net
Step 2: login in with given password: 6dd28e9b
Step 3: explore file directory with commands like: ls
Step 4: verify integrity of file use command: sha256sum
Step 5: use cat checksum.txt to read and display the content of the files
Step 6: decrypt the file hash

```
ctf-player@pico-chall$ cat checksum.txt
03b52eabed517324828b9e09cbbf8a7b0911f348f76cf989ba6d51acede6d5d8
ctf-player@pico-chall$ ./decrypt.sh files/00011a60
picoCTF{trust_but_verify_00011a60}
ctf-player@pico-chall$ Connection to rhea.picoctf.net closed by remote host.
Connection to rhea.picoctf.net closed.
```

FLAG: picoCTF{trust_but_varify_00011a60}

Hashcrack
Easy, Cryptography, picoCTF 2025, browser_webshell_solvable

Descritption: A company stored a secret message on a server which got breached due to the admin using weakly hashed passwords. Can you gain access to the secret stored within the server?
Access the server using nc verbal-sleep.picoctf.net 57271

Hint:
-Understanding hashes is very crucial. Read more here.
-Can you identify the hash algorithm? Look carefully at the length and structure of each hash identified.
-Tried using any hash cracking tools?

Hashing cracking challenge. Tools to help me solve these challenges would include hashcat, python and hashlib. First hashcode was recognized at MD5 and second as SHA-1 and thrid as SHA-3. After third hash passcode was enter Flag was revealed

```
q71tar-picoctf@webshell:~$ nc verbal-sleep.picoctf.net 57271
Welcome!! Looking For the Secret?

We have identified a hash: 482c811da5d5b4bc6d497ffa98491e38
Enter the password for identified hash: password123
Correct! You've cracked the MD5 hash with no secret found!

Flag is yet to be revealed!! Crack this hash: b7a875fc1ea228b9061041b7cec4bd3c52ab3ce3
Enter the password for the identified hash: letmein
Correct! You've cracked the SHA-1 hash with no secret found!

Almost there!! Crack this hash: 916e8c4f79b25028c9e467f1eb8eee6d6bbdff965f9928310ad30a
8d88697745
Enter the password for the identified hash: qwerty098
Correct! You've cracked the SHA-256 hash with a secret found.
The flag is: picoCTF{UseStr0nG_h@shEs_&PaSswDs!_eff9dbe0}
```

W